(12) UK Patent Application (19) GB (11) 2 277 002 (13) A

(43) Date of A Publication 12.10.1994

(21) Application No 9307442.5

(22) Date of Filing 08.04.1993

(71) Applicant(s)
Sony United Kingdom Limited

(Incorporated in the United Kingdom)

Sony House, South Street, STAINES, Middlesex,
TW18 4PF, United Kingdom

(72) Inventor(s)
Martin Rex Dorricott
Morgan William Amos David
Shima Ravji Varsani

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA,
United Kingdom

(51) INT CL^5
H04N 5/14 , G06F 15/70

(52) UK CL (Edition M)
H4F FD1B9 FD10 FD12X FD3 FD30J FD30K FD30R FD54 FGM

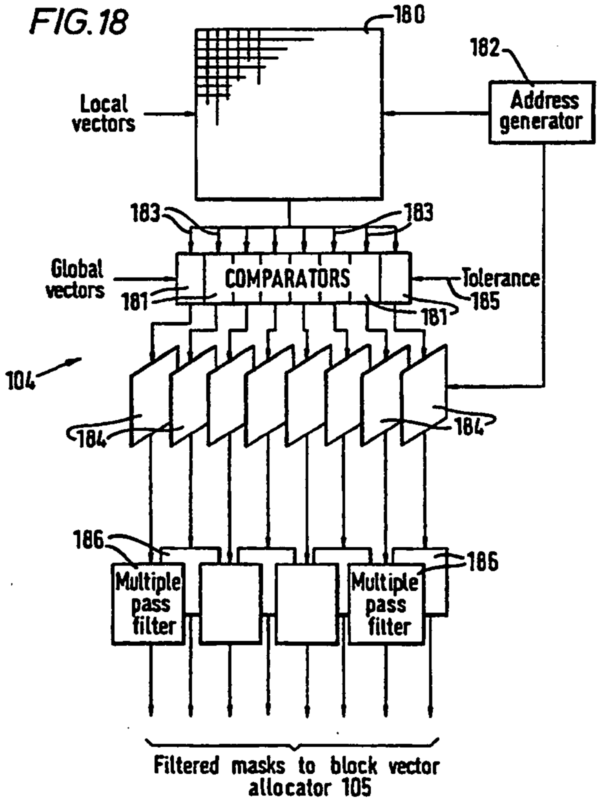
(56) Documents Cited
GB 2249899 A EP 0498938 A2

(58) Field of Search
UK CL (Edition L) H4F FGM
INT CL^5 G06F 15/70 , H04N 5/14
Online databases: WPI, INSPEC.

(54) Selecting motion vectors in video signal processing; global motion vectors.

(57) An image is divided into blocks and local motion vectors are derived for each block indicating the possible motions of the block between frames. Global motion vectors are derived by selecting the most frequently occurring local vectors over the whole image (103, fig 13). Association means 104 determine that a given global motion vector is "associated" with a given block by comparing 181 it with the local motion vectors. A global motion vector is associated with a block if it lies within a predetermined threshold 185 of the local motion vectors for that block. Assignment of motion vectors in a block (105, fig 13) is then made on a pixel by pixel basis from: the local motion vectors for that block, the local motion vectors for neighbouring blocks, and those global vectors (if any) associated with that block.

FIG.18



GB 2 277 002 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

FIG. 1

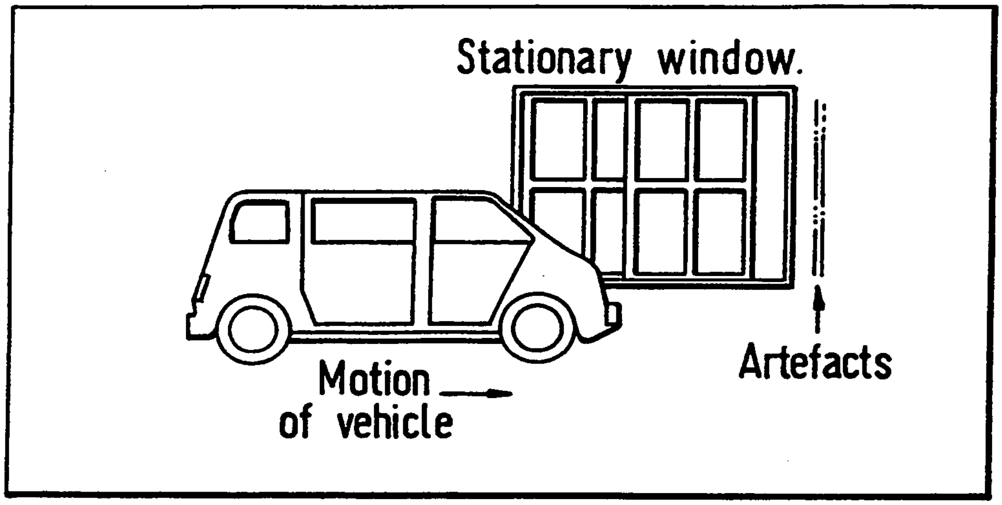

Stationary window.
Artefacts
Motion of vehicle

FIG. 19

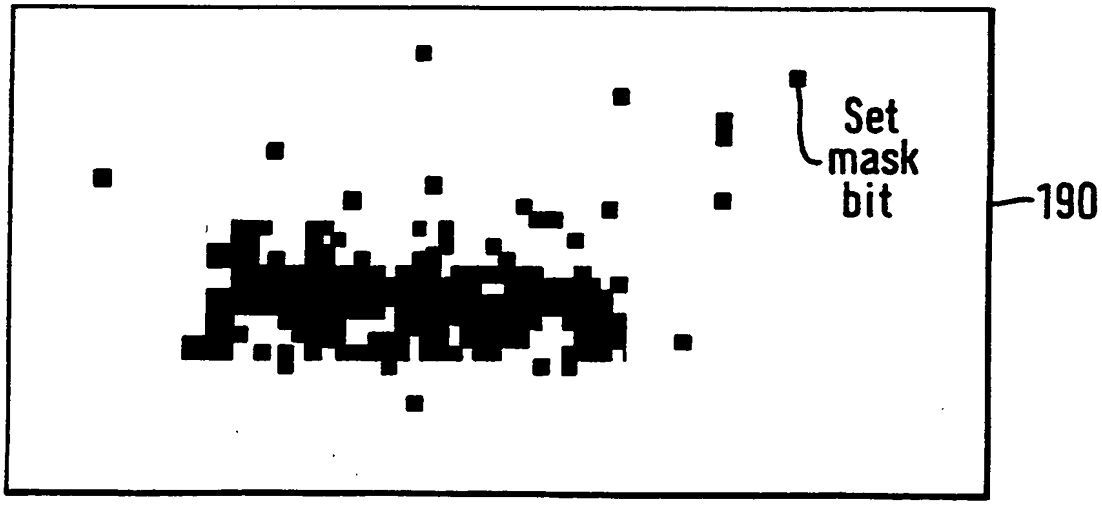

Set mask bit
190

FIG. 2

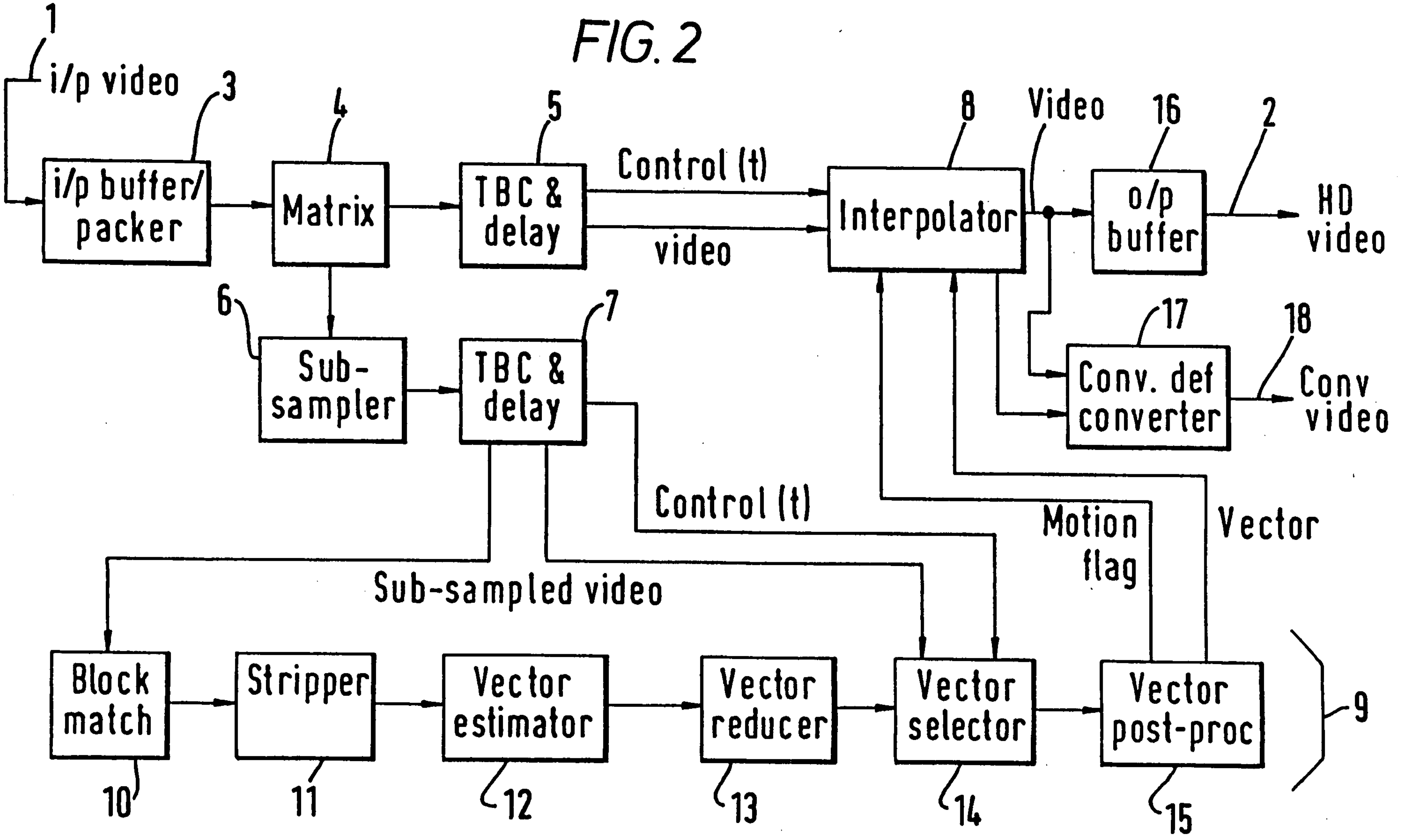

1
i/p video
3
i/p buffer/ packer
4
Matrix
5
TBC & delay
Control (t)
video
8
Interpolator
Video
16
o/p buffer
2
HD video
6
Sub-sampler
7
TBC & delay
17
Conv. def converter
18
Conv video
Control (t)
Motion flag
Vector
Sub-sampled video
Block match
10
Stripper
11
Vector estimator
12
Vector reducer
13
Vector selector
14
Vector post-proc
15
9

FIG. 3

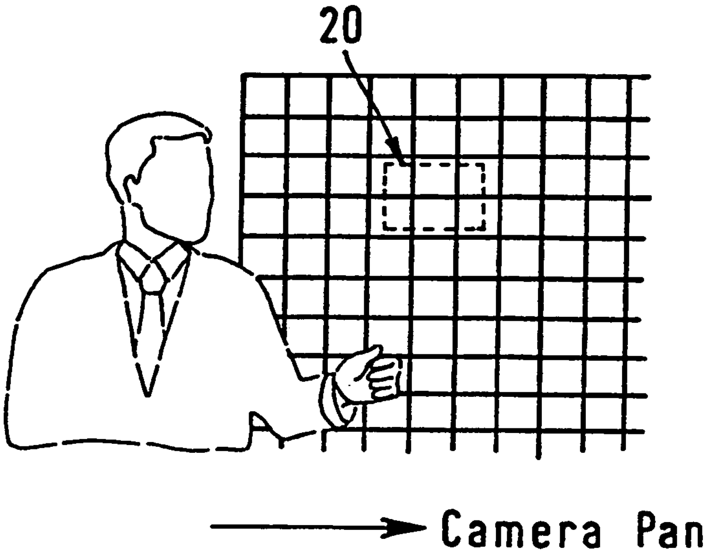

20
Camera Pan

FIG. 4

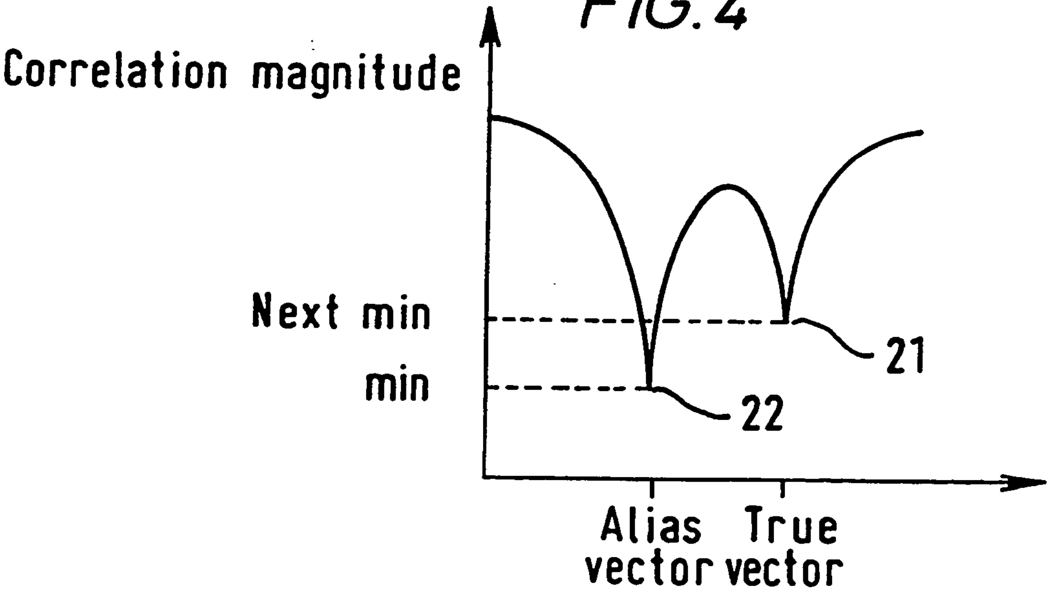

Correlation magnitude
Next min
min
21
22
Alias vector
True vector

FIG.5

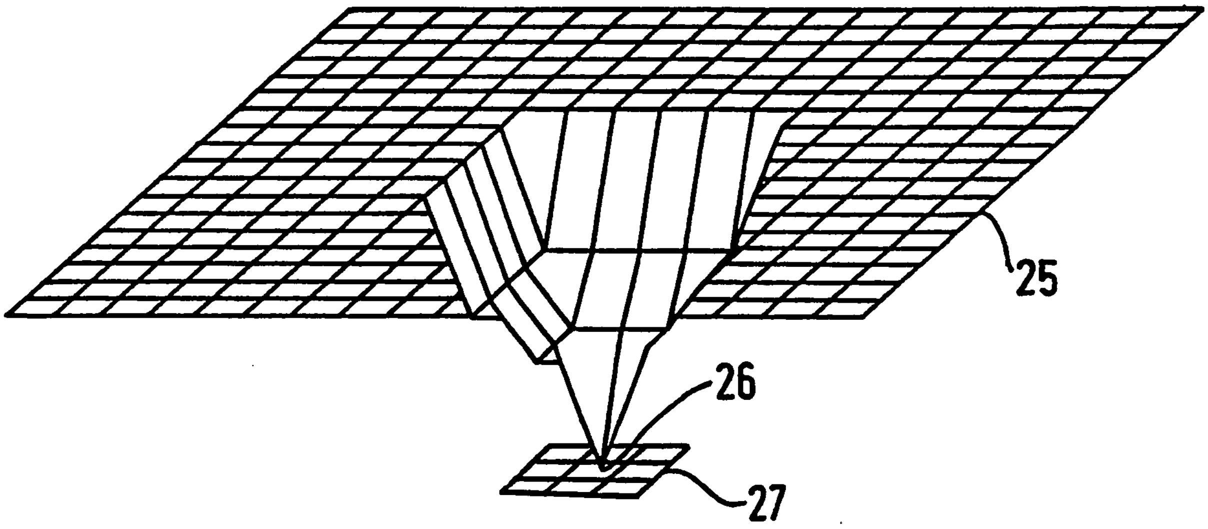


FIG.6a

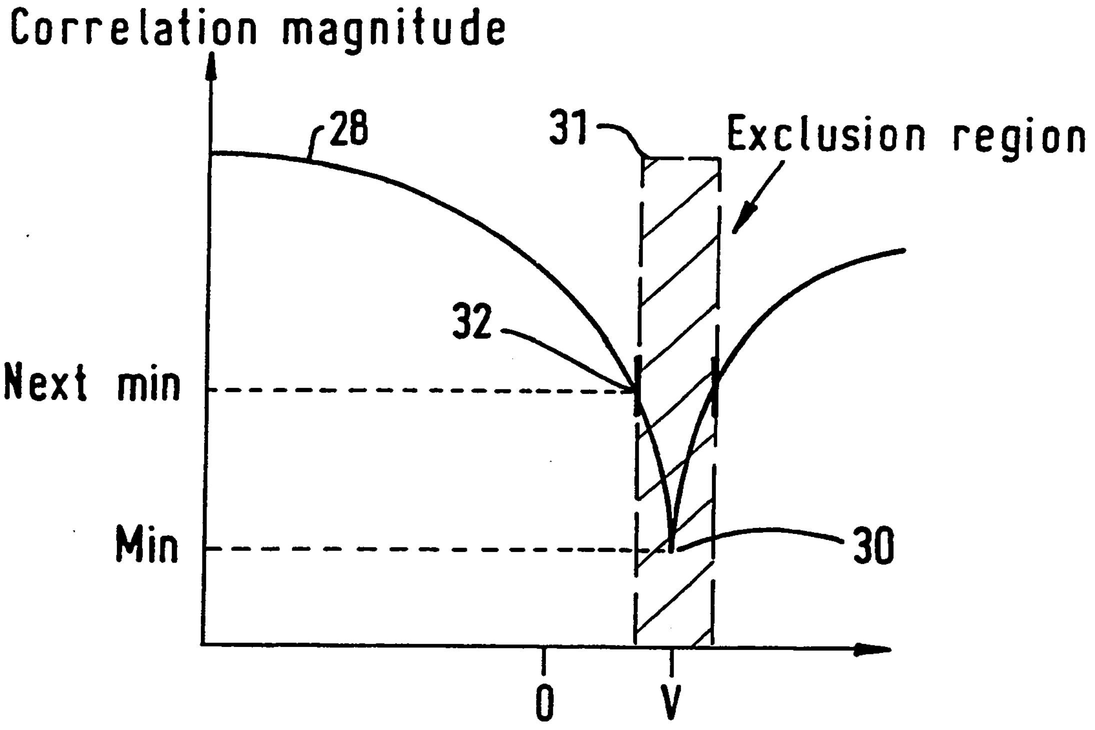

Correlation magnitude
28
31
Exclusion region
32
Next min
Min
30
0
V

FIG.6b

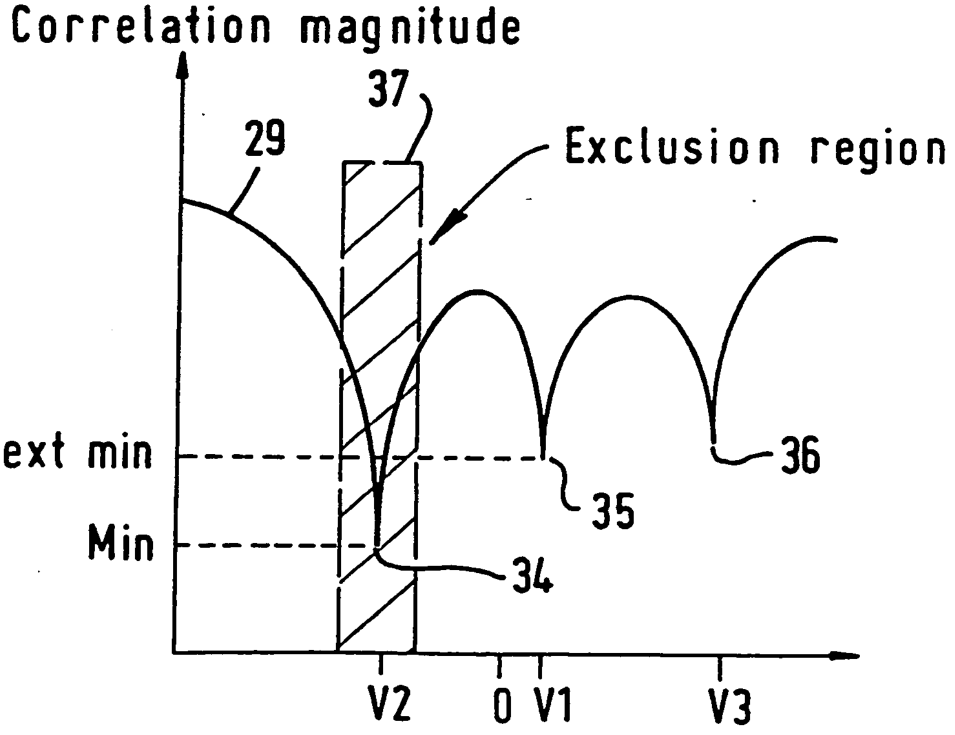

Correlation magnitude
37
29
Exclusion region
Next min
Min
35
36
34
V2
0
V1
V3

FIG. 7

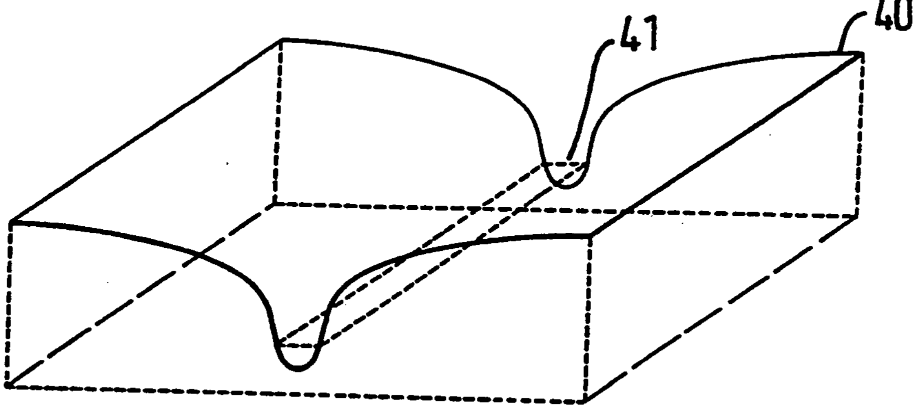


FIG.11

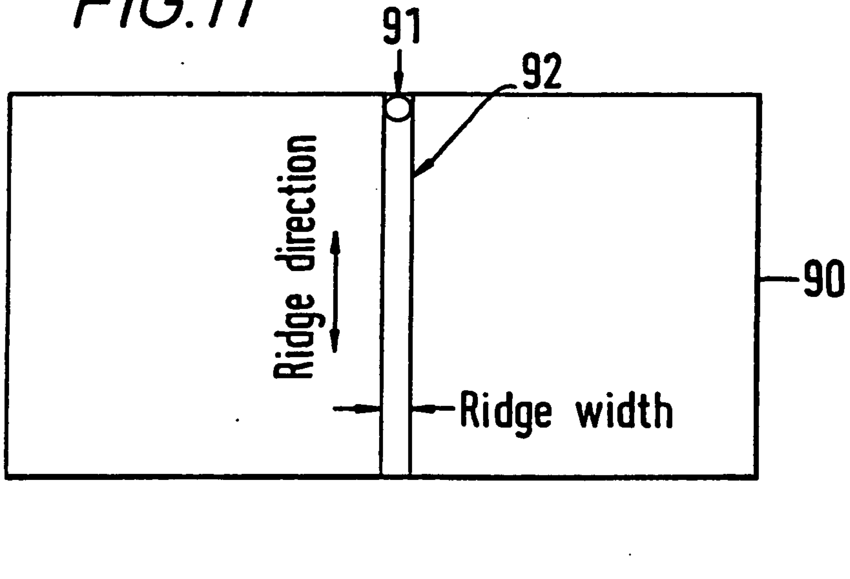
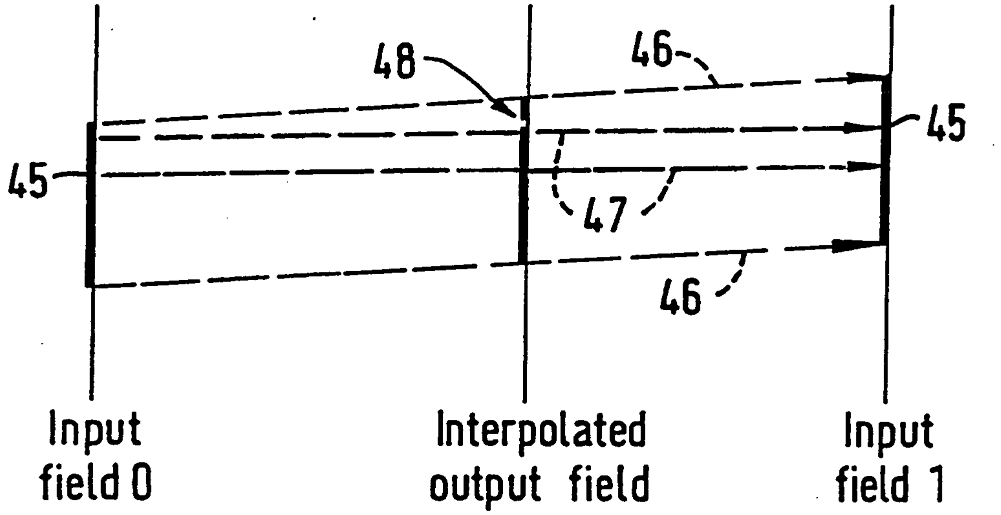


FIG. 8



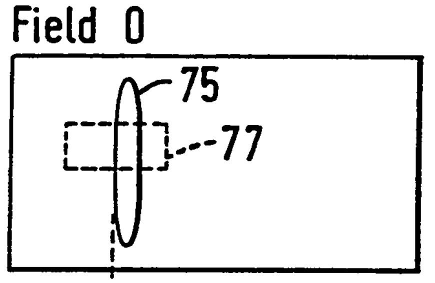


FIG. 10a

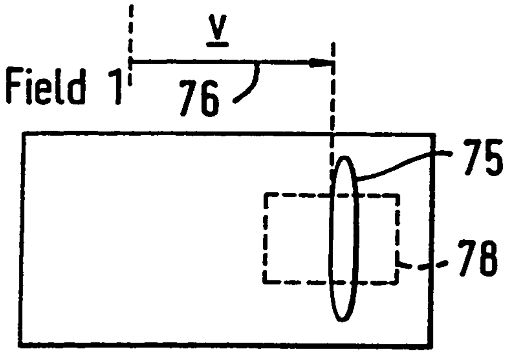


FIG. 10b

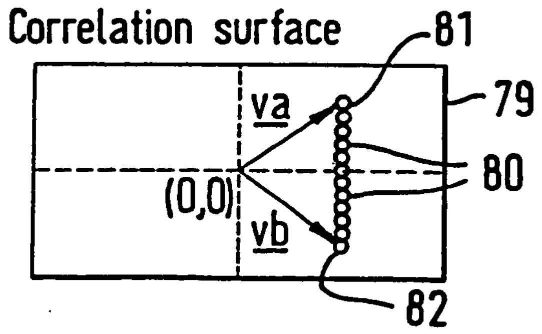


FIG. 10c

FIG. 9

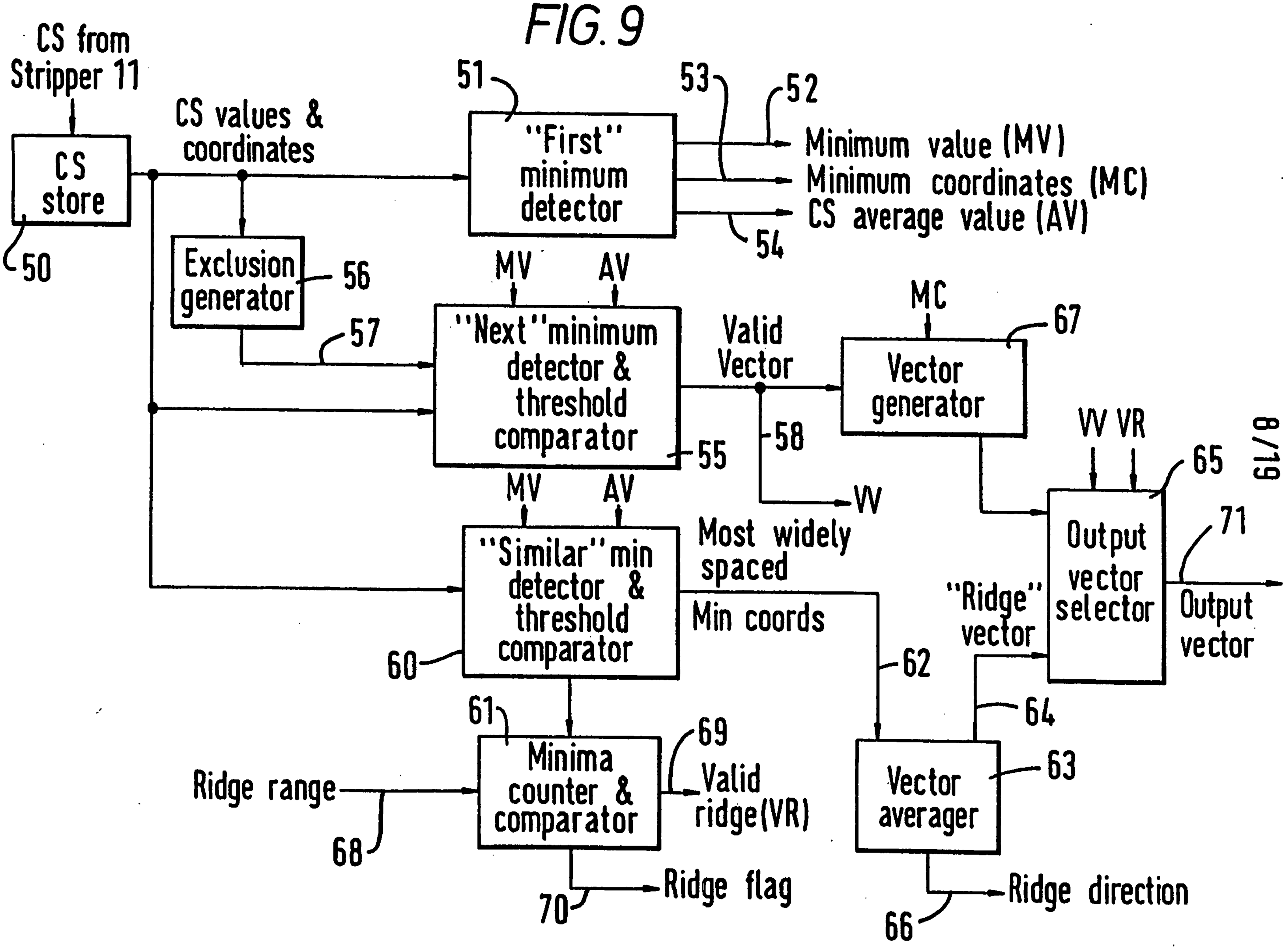

CS from
Stripper 11
CS
store
50
CS values &
coordinates
Exclusion
generator
56
57
51
"First"
minimum
detector
53
52
54
Minimum value (MV)
Minimum coordinates (MC)
CS average value (AV)
MV
AV
"Next" minimum
detector &
threshold
comparator
55
Valid
Vector
58
W
MC
Vector
generator
67
VV VR
65
Output
vector
selector
71
Output
vector
MV
AV
"Similar" min
detector &
threshold
comparator
60
Most widely
spaced
Min coords
62
"Ridge"
vector
64
Vector
averager
63
Ridge direction
66
61
Minima
counter &
comparator
Ridge range
68
69
Valid
ridge (VR)
Ridge flag
70

FIG.12a

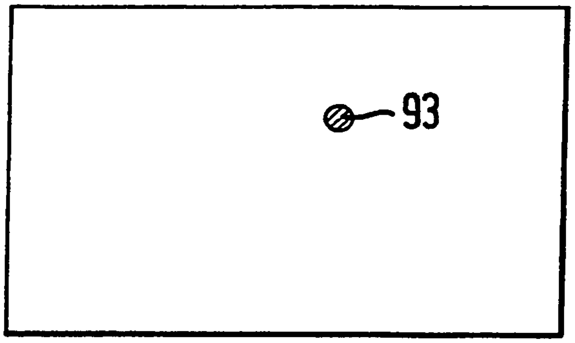


FIG.12b

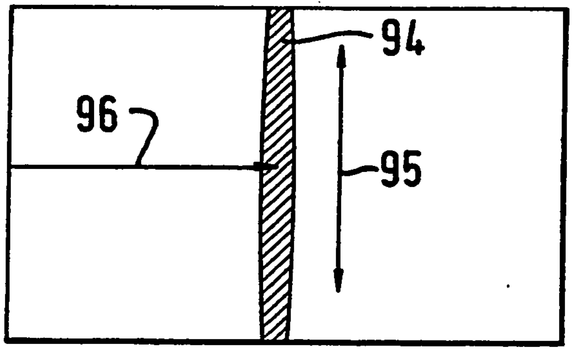
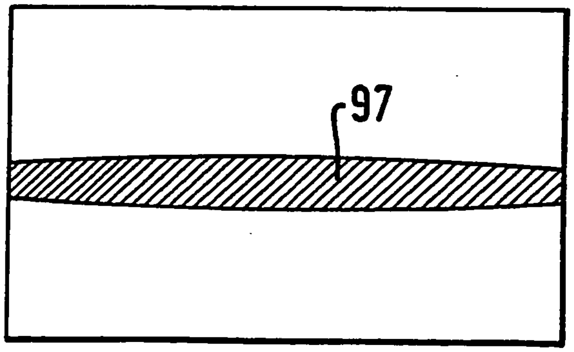


FIG.12c



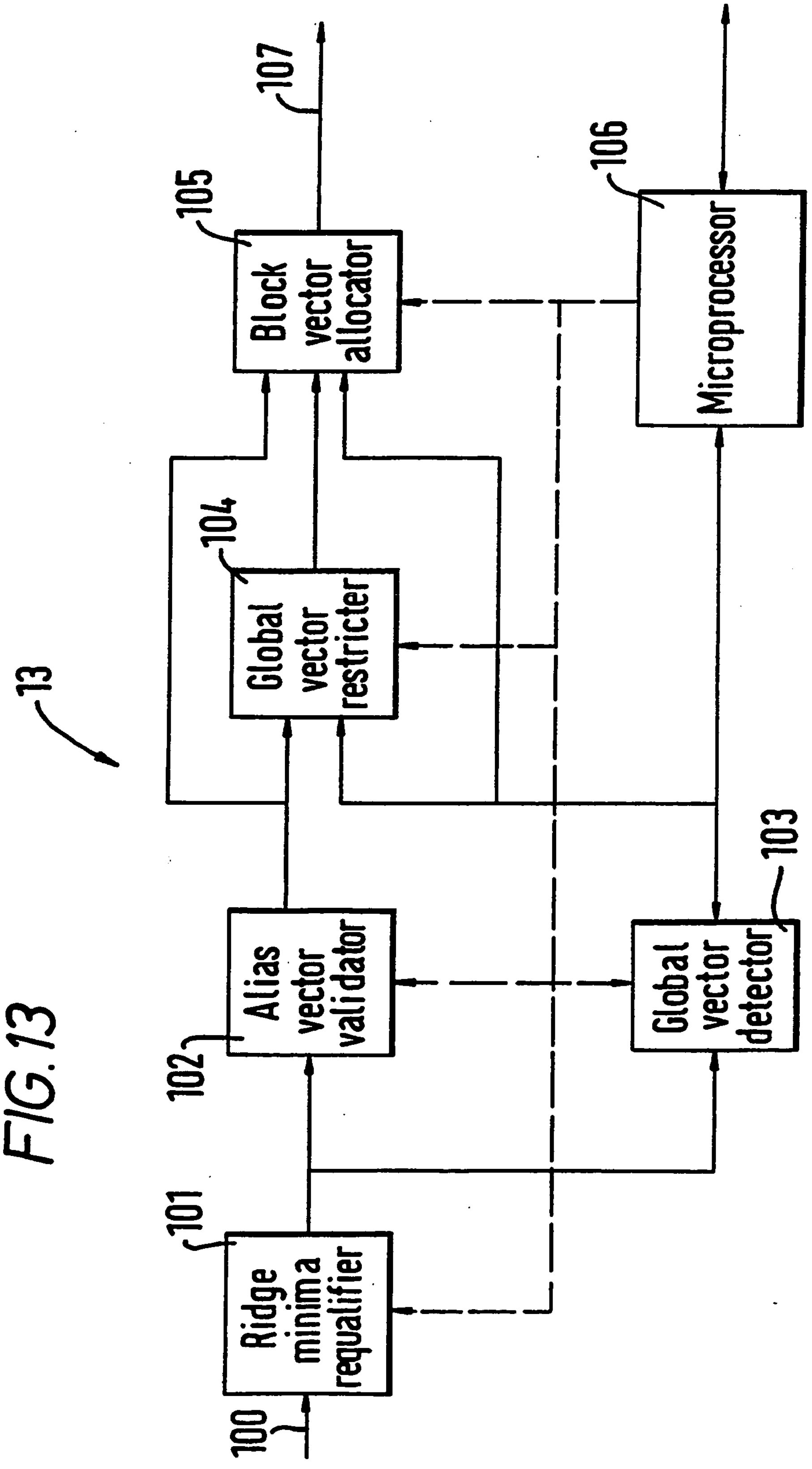

13
100
101
Ridge minima requalifier
102
Alias vector validator
103
Global vector detector
104
Global vector restricter
105
Block vector allocator
106
Microprocessor
107

FIG.13

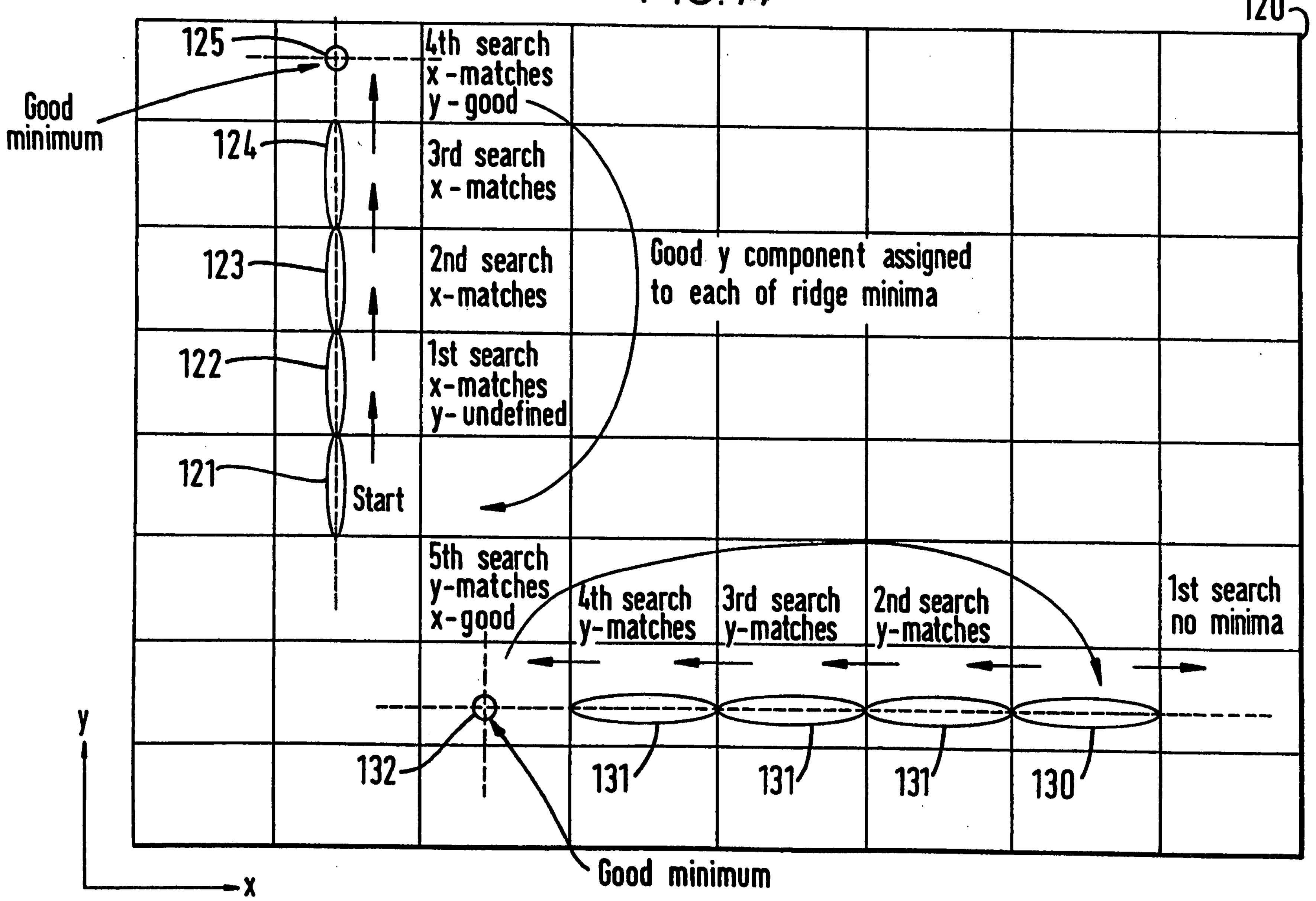

120
125
Good
minimum
4th search
x -matches
y - good
124
3rd search
x - matches
123
2nd search
x-matches
122
1st search
x-matches
y- undefined
121
Start
Good y component assigned
to each of ridge minima
5th search
y-matches
x-good
4th search
y-matches
3rd search
y-matches
2nd search
y-matches
1st search
no minima
132
131
131
131
130
Good minimum
y
x

FIG. 14

FIG. 15

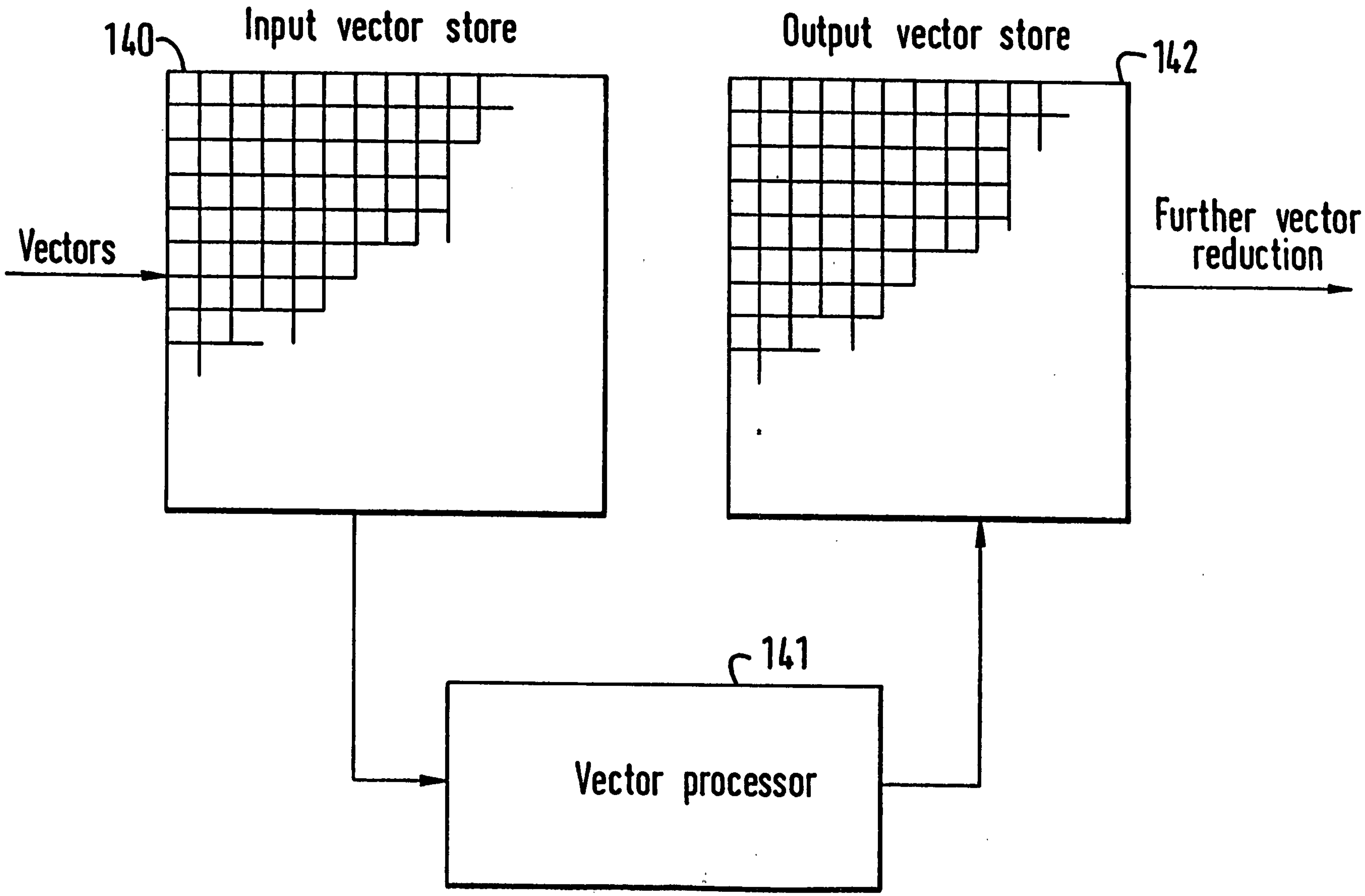

140
Input vector store
Output vector store
142
Vectors
Further vector reduction
141
Vector processor

FIG. 16

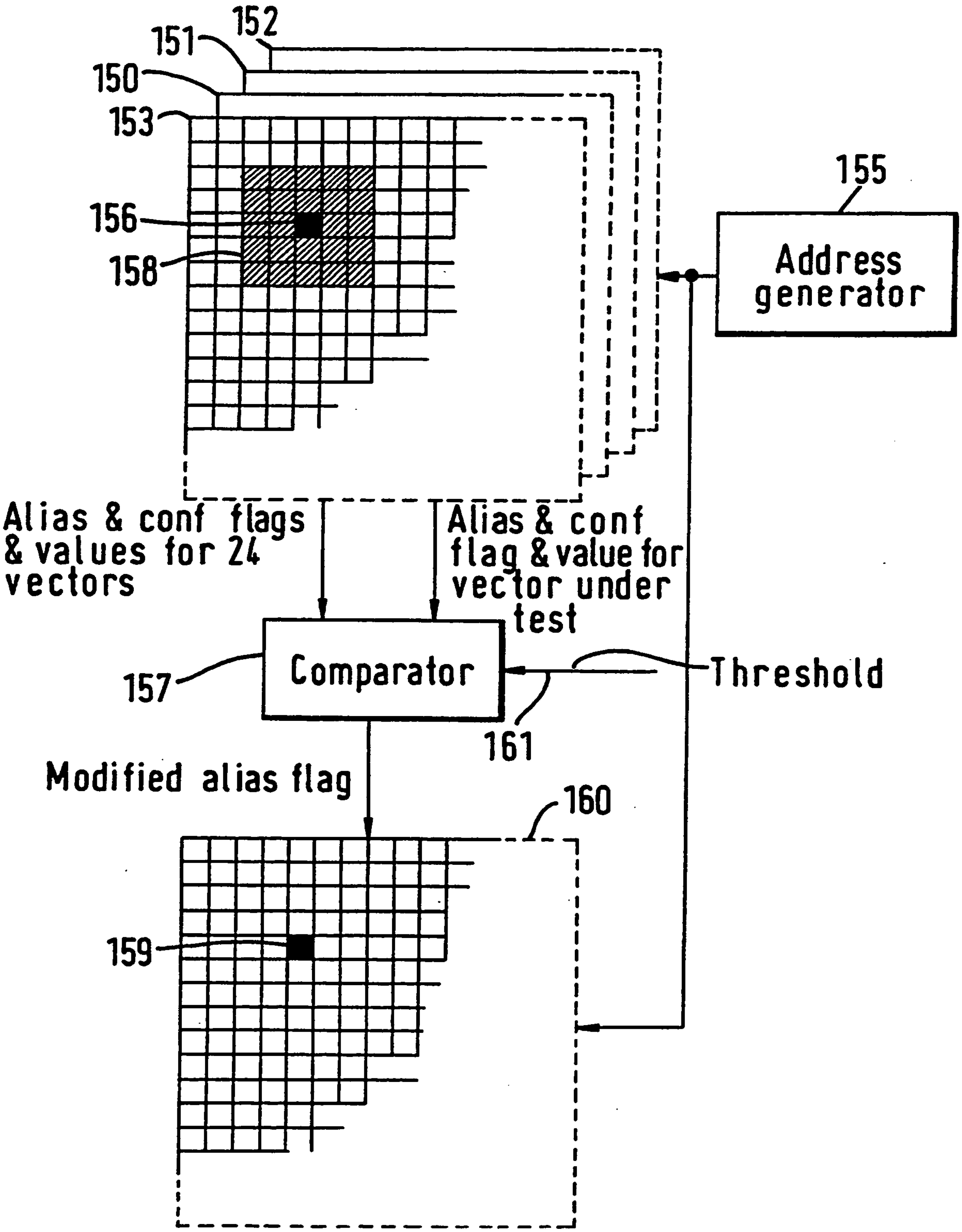

152
151
150
153
156
158
155
Address generator
Alias & conf flags & values for 24 vectors
Alias & conf flag & value for vector under test
Comparator
157
Threshold
161
Modified alias flag
160
159

FIG. 17

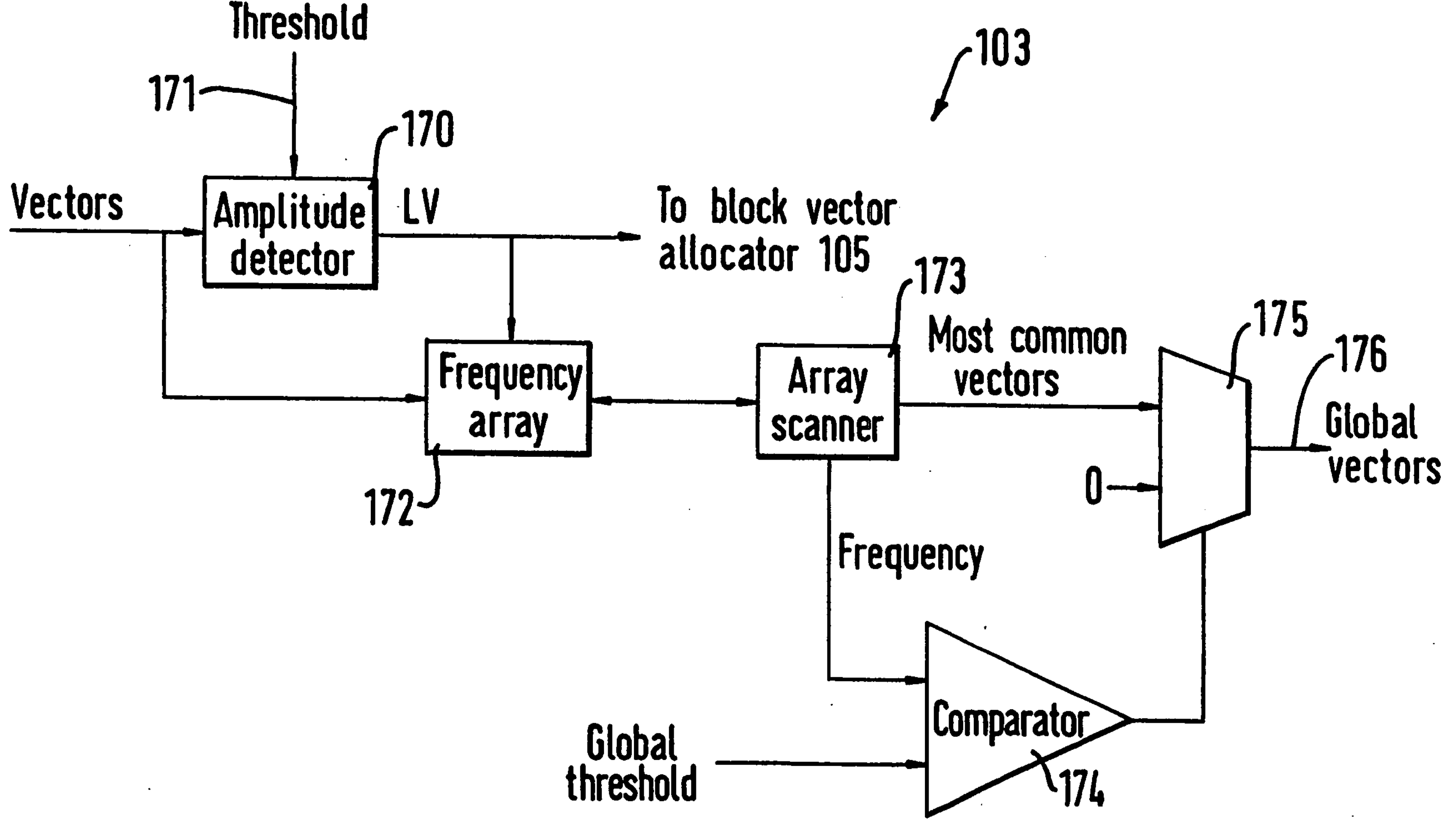
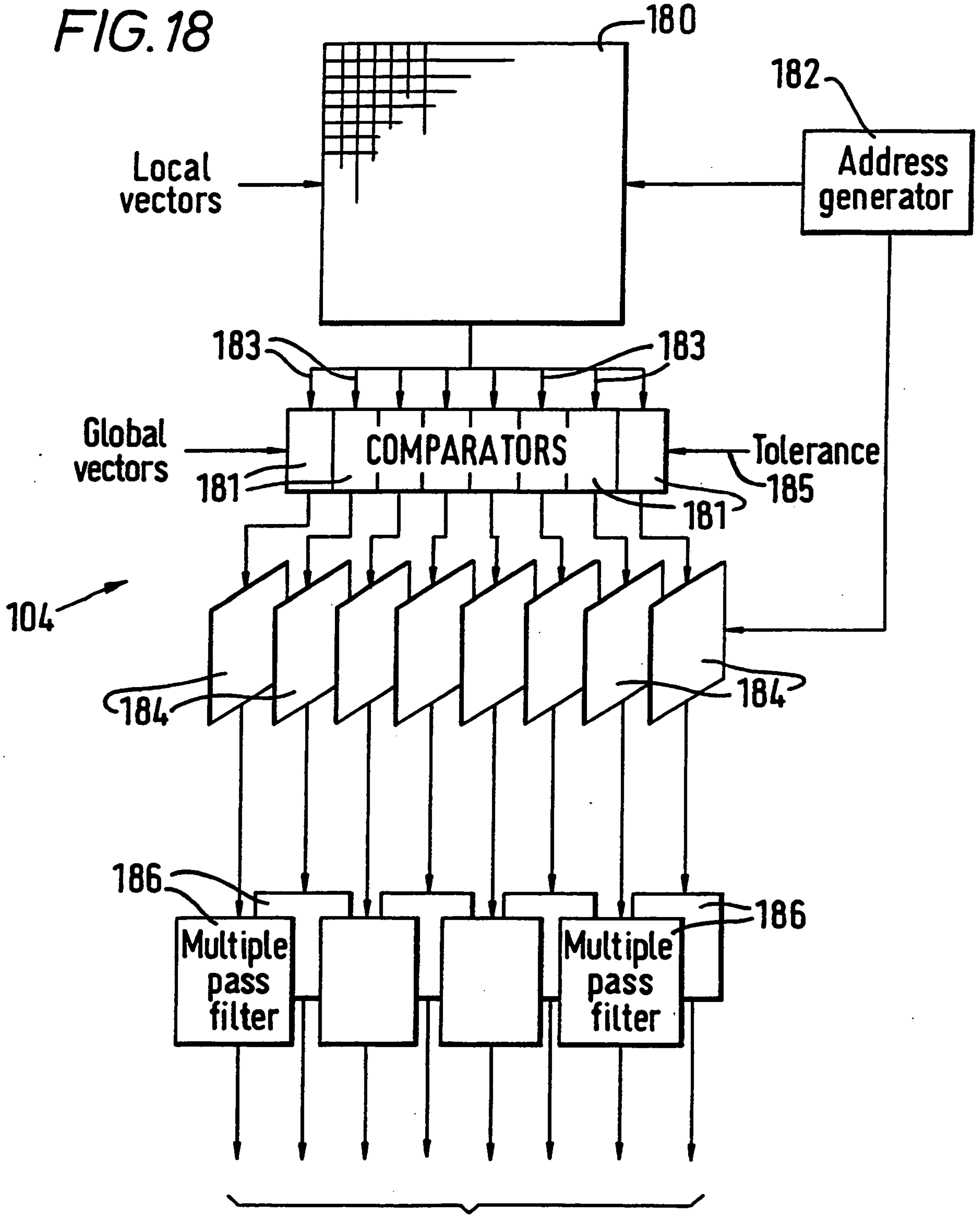

Threshold
171
170
Vectors
Amplitude detector
LV
To block vector allocator 105
103
173
Most common vectors
175
176
Global vectors
Frequency array
172
Array scanner
0
Frequency
Global threshold
Comparator
174

FIG. 18



Filtered masks to block vector allocator 105

FIG. 20

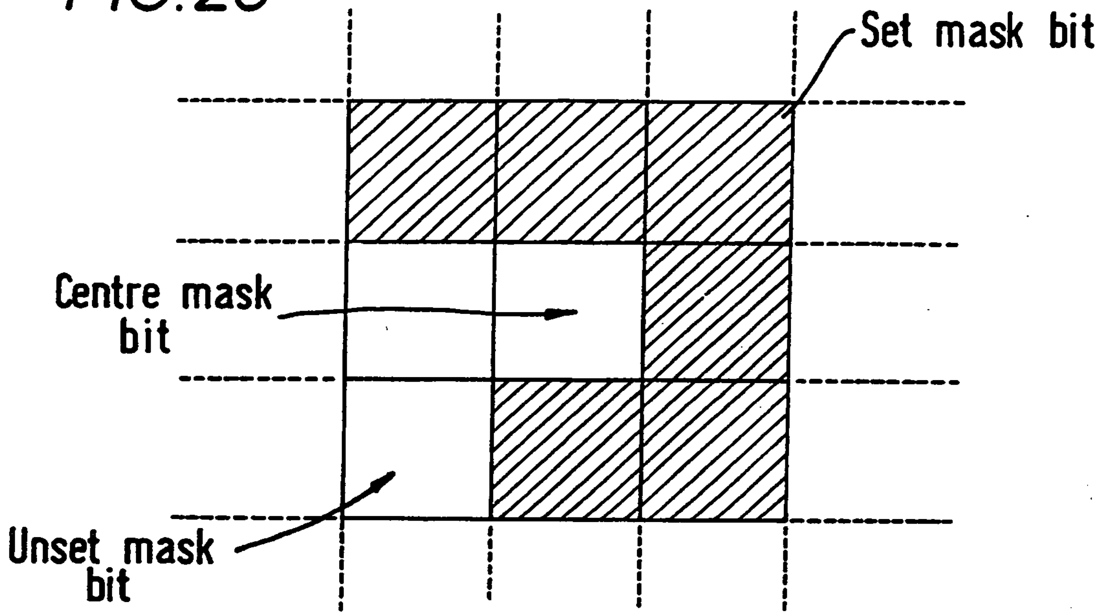


FIG. 21

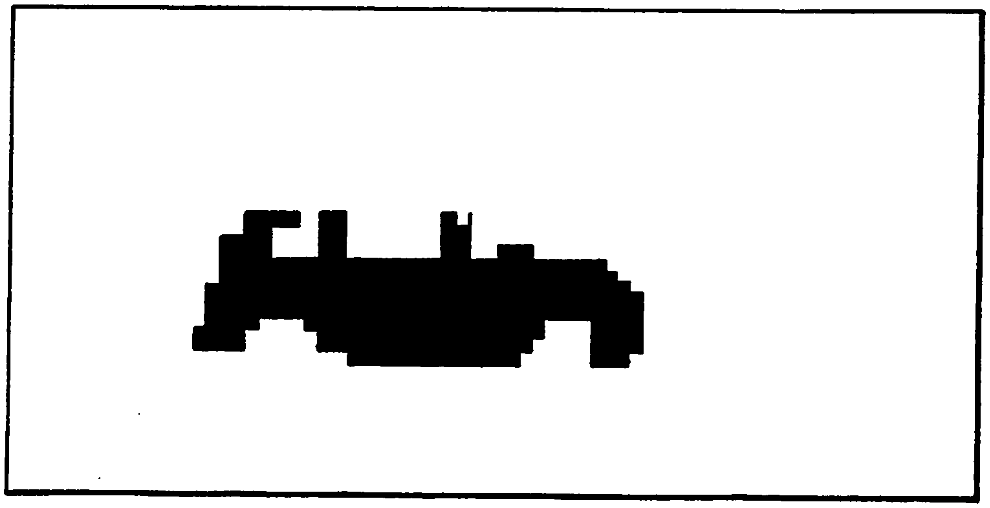


FIG. 22 (1/2)

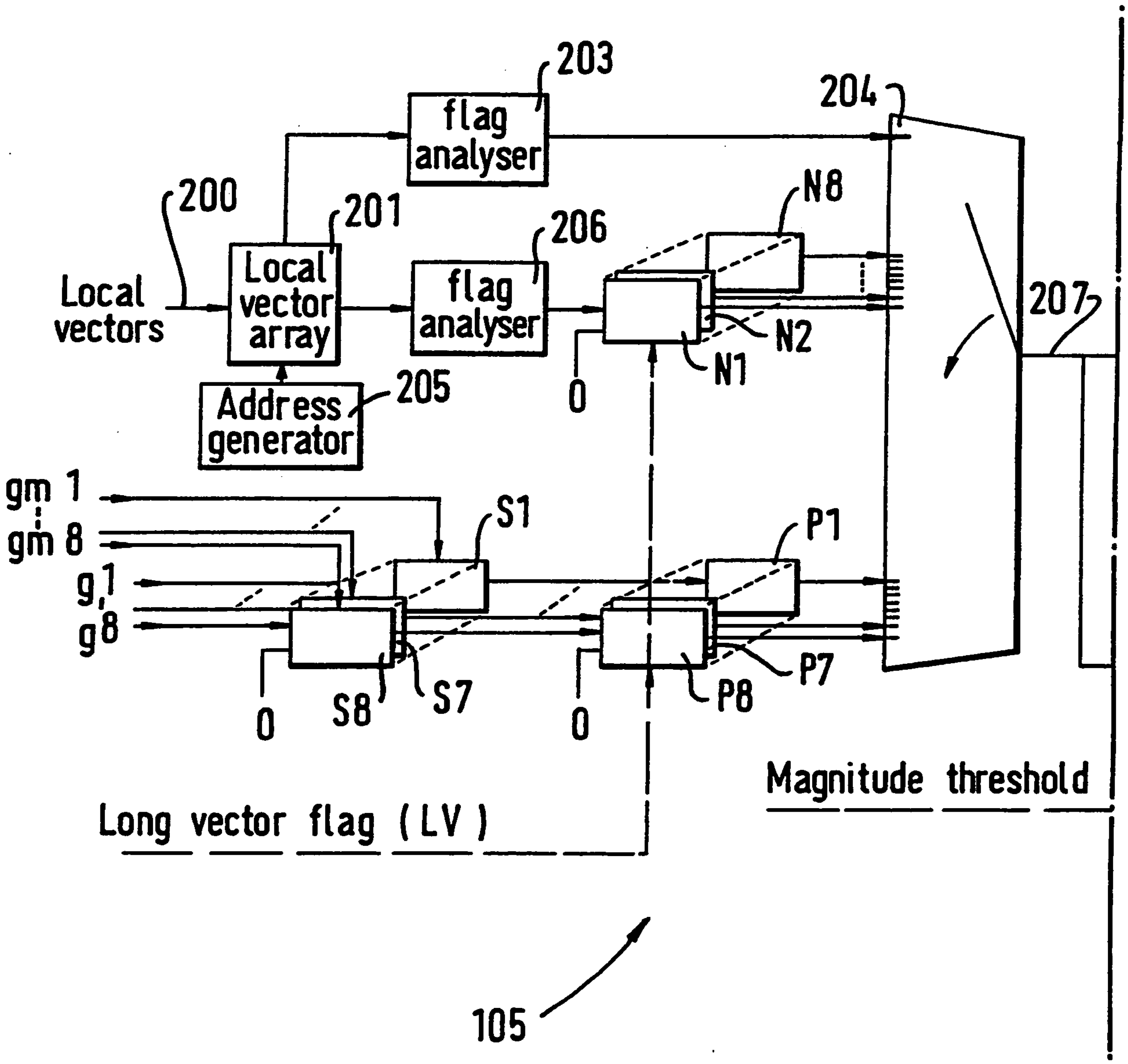

203
flag analyser
204
200
201
N8
206
Local vectors
Local vector array
flag analyser
207
N2
N1
0
Address generator
205
gm 1
gm 8
g 1
g 8
S1
P1
P7
S8
S7
P8
0
0
Magnitude threshold
Long vector flag (LV)
105

FIG. 22 (2/2)

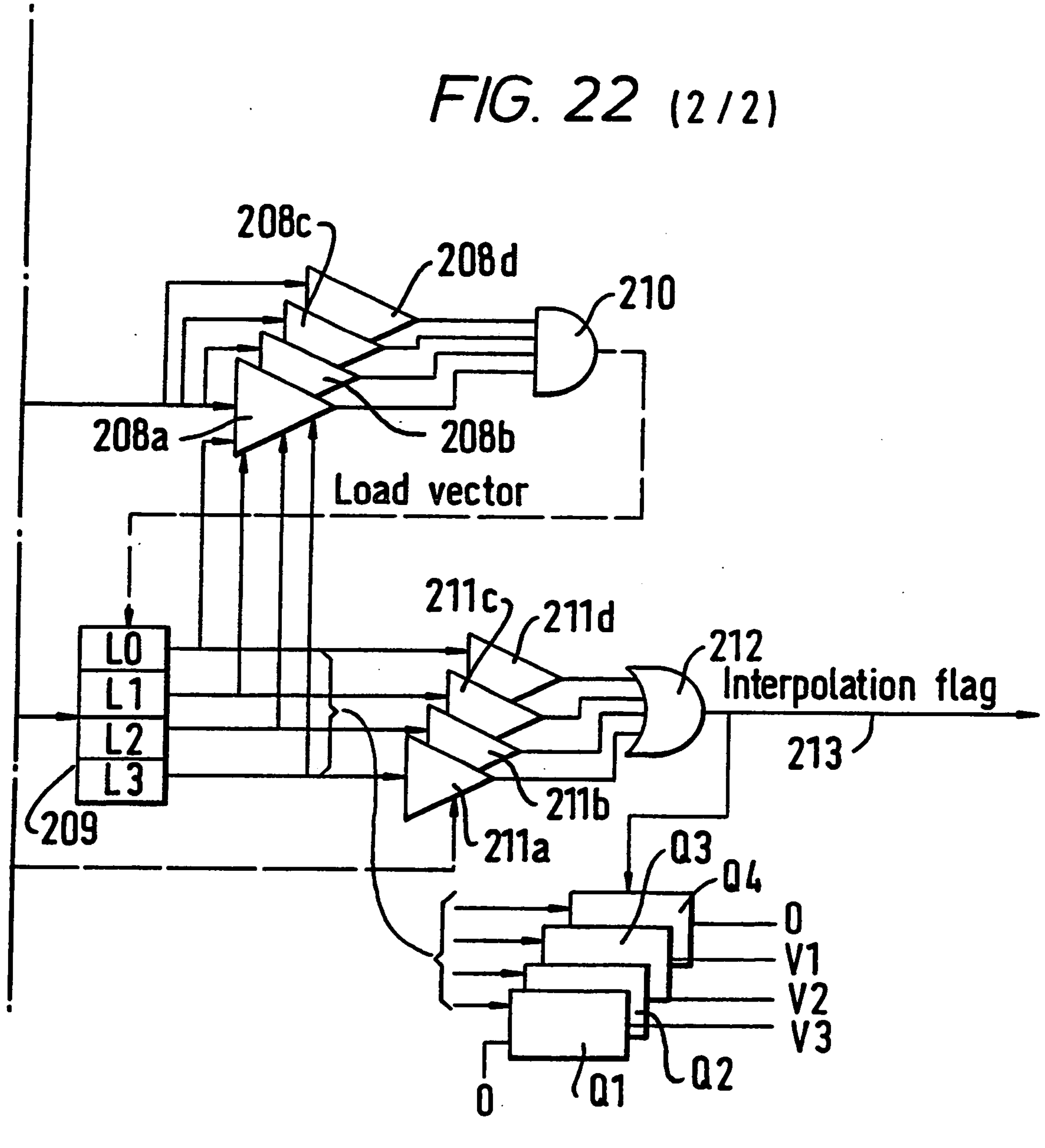
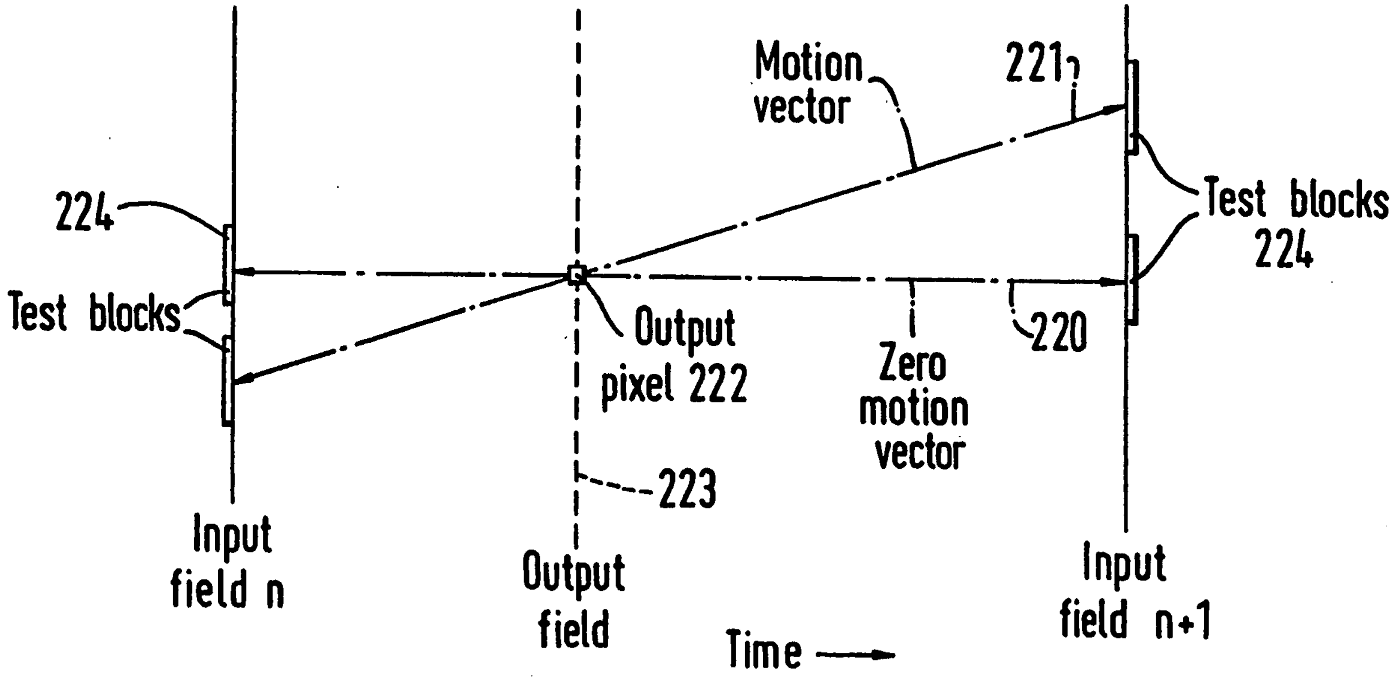

208c
208d
210
208a
208b
Load vector
211c
211d
212
Interpolation flag
213
L0
L1
L2
L3
209
211b
211a
Q3
Q4
0
V1
V2
V3
Q2
Q1
0

FIG. 23


Motion vector
221
224
Test blocks
Test blocks 224
Output pixel 222
Zero motion vector
220
223
Input field n
Output field
Time
Input field n+1

MOTION COMPENSATED VIDEO SIGNAL PROCESSING

This invention relates to motion compensated video signal processing.

Motion compensated video signal processing is used in applications such as television standards conversion, film standards conversion and conversion between video and film standards.

In a motion compensated television standards converter, such as the converter described in UK patent application number GB-A-2 231 749, pairs of successive input images are processed to generate sets of motion vectors representing image motion between the pair of input images. The processing is carried out on discrete blocks of the images, so that each motion vector represents the inter-image motion of the content of a respective block.

In the motion vector estimation process a correlation surface representing spatial correlation between blocks of the two input images is examined to detect points of maximum correlation. (The correlation surface actually represents the difference between the two input images so that the points of maximum correlation are in fact minima on the correlation surface, and will be referred to as such herein.) When a minimum has been detected, a motion vector is generated from the spatial position of the minimum in the correlation surface. Tests are performed to determine whether that minimum represents a significant peak in correlation with respect to the remainder of the correlation surface. If the minimum passes this test it is treated as being "valid", and a confidence flag associated with the motion vector is set.

After the individual motion vectors, termed "local" motion vectors, have been derived for the blocks of the input image, the local motion vectors are supplied to a motion vector reducer. The motion vector reducer assigns to each block a set of motion vectors selected, in order of preference, from: the zero motion vector; the local motion vector for that block; the local motion vectors for blocks neighbouring that block in the input image; and a set of vectors known as "global" motion vectors. The global motion vectors are derived for each image by ranking all the (valid) local motion vectors for this image in order of decreasing frequency, and then selecting from the most common of the local motion vectors a number of unique motion vectors to be used as global motion vectors. A constraint may be applied here to ensure that the global motion vectors differ from each other by at least a minimum amount.

The sets of motion vectors for all the blocks of the input image are then supplied to a motion vector selector. The purpose of the motion vector selector is to allocate to each picture element (pixel) of a block of an output image a single motion vector selected from the set of motion vectors supplied for a corresponding block of the input image. As this motion vector will be used in interpolation of the output pixel, the motion vectors supplied to the motion vector selector are tested so that the "best" motion vector is selected for each pixel from the set of motion vectors for the corresponding block. The selected motion vector for each pixel is then supplied to a motion compensated interpolator which uses these motion vectors to interpolate the pixels of an output image from the pair of input images in dependence upon the temporal offset of the output image from the pair of input images.

The purpose of the motion vector reducer is therefore to provide the motion vector selector with a reasonable selection of motion vectors for each block from which to select the most appropriate motion vector for each output pixel corresponding to that block. The global motion vectors each represent a motion which was frequently detected in the image as a whole, so that it is often reasonable to include these vectors in the sets of vectors from which the most appropriate vectors are selected by the motion vector selector. However, the process of calculating the global vectors from an image as a whole and then making them available for use over the whole area of the image means that vectors derived from only part of the image may be selected for pixels corresponding to an entirely different part of the image. This can result in disturbing artifacts in the output image as illustrated in Figure 1 of the accompanying drawings. Here, a global motion vector corresponding to the motion of a vehicle has been inappropriately selected for use in interpolation of the region of the image to the right of a (stationary) window in the background. The result is the double-imaging of the right-hand vertical edge of the window as shown in the figure.

UK patent application number GB-A-2249899 discloses a modification to the above technique wherein the input image is considered to be sub-divided into an array of rectangular areas and global vectors are derived for each area. Thus, the global vectors for each area are derived by considering the local vectors associated with blocks in that area. In the motion vector reducer, only the global vectors derived for a given area may be assigned to blocks within that area. While this technique attempts to restrict the application of global vectors to some extent, it is likely that the input images contain several features moving in different ways and any one of these features could overlap a number of areas of the rectangular array. Thus, depending on the picture content, a different global vector could be derived for each of the areas overlapped by a single feature, with the result that inappropriate global vectors may still be selected for use in the interpolation process. An effect of this can be to produce a visible block effect in the output image corresponding to the artificial division of the original input image into the array of rectangular areas.

According to the present invention there is provided motion compensated video signal processing apparatus in which motion vectors are generated to represent image motion between a pair of images of an input video signal, the apparatus comprising:

means for deriving local motion vectors representing the motion of the content of respective blocks of one image of the pair between that image and the other image of the pair;

means for deriving from the local motion vectors a set of global motion vectors comprising a plurality of unique motion vectors selected from the most common of the local motion vectors;

association means for establishing an association between each global vector and blocks of the said image, the association means comprising means for comparing each global motion vector with each local motion vector, the said association being established between a global motion vector and a given block if the global motion vector is within a predetermined tolerance of the local motion vector for that block; and

means for assigning to each said block a plurality of motion vectors selected from the zero motion vector, the local motion vector for that block, the local motion vectors for the blocks neighbouring that block in the input image, and any global motion vectors for which the said association with that block has been established.

In accordance with this invention, therefore, global vectors are derived from the input image as a whole, but the application of global vectors is restricted to parts of the image at or near areas in which motion similar to that represented by each global motion vector was initially detected. Each global motion vector is compared with the local motion vectors derived for the blocks of the input image, and an association is established between the global motion vector and any block of the image for which the corresponding local motion vector is identical or similar to the global motion vector. Only global motion vectors for which the said association has been established with a given block can be assigned to that block for supply to the motion vector selector.

The said association means conveniently include means for generating for each global vector a global mask array having an array entry for each said block of the input image, wherein each array entry indicates whether or not the said association has been established for the corresponding block.

If the said association between a global motion vector and blocks of the input image is established solely on the basis of a comparison of the global motion vector and the local motion vectors, then spurious points of association may occur where, for example, noise in the input signal results in an incorrect, but apparently valid, local motion vector being similar to a global motion vector. Equally, the association may fail to be established through incorrect local motion vectors which fail the global comparison test. To eliminate such spurious errors, it is preferred that the association means includes first-pass mask adjustment means for determining, for each entry in each global mask array, the number of entries which indicate that the said association has been established in a group of entries comprising that entry and the entries corresponding to the blocks neighbouring the block to which that entry corresponds, comparing the said number with a predetermined threshold, and, if the number is less than the threshold and the entry indicates that the said association has been established, resetting the entry to indicate that the association has not been established, and, if the number is greater than or equal to the threshold and the entry indicates that the said association has not been established, resetting the entry to indicate that the association has been established. This first-pass mask adjustment process is performed over the whole of each global mask array by considering only the original, unadjusted array entries generated through comparison of the global motion vector and the local motion vectors. That is to say, during the first-pass process, the adjustment is not taken into account in any further consideration of that entry as the first-pass adjustment process continues. This can be achieved, for example, by reading the original, unadjusted mask array entries from a first array memory and writing the adjusted mask array entries to a different array memory. Alternatively, an appropriate delay can be effected in the mask adjustment process to prevent the resetting of entries affecting the remainder of the first-pass process.

This first-pass mask adjustment process is effectively a filtering process which eliminates spurious points of association in the global mask arrays since the entries corresponding to blocks neighbouring a block for which a spurious association is indicated will indicate no association, and the array entry for that block will be reset accordingly to indicate no association. Equally, spurious points which erroneously indicate no association for a given block will be reset to indicate an association if a sufficient number of the entries corresponding to neighbouring blocks indicate that the association has been established.

While the first-pass mask adjustment means serves to eliminate spurious errors in the global mask arrays, it may also have the effect of "shrinking" the areas in each global mask array over which the said association was initially correctly established. That is to say, the entries corresponding to blocks at the edges of a solid moving object in the input image may be reset to indicate no association since the entries corresponding to blocks beyond the edge of the moving object are considered in the first-pass mask adjustment. Accordingly, it is preferred that the association means includes additional-pass mask adjustment means for repeating the operation of the first-pass mask adjustment means at least once with the said predetermined threshold set to a lower level than that for the first-pass mask adjustment.

Since the threshold is set to a lower level, the effect of the mask adjustment process after the first pass is to cause "spreading" of the areas in each global mask array over which the said association is established. This reverses the "shrinkage" introduced by the first-pass mask adjustment. However, to obtain a resultant increase in the areas over which the association is established, it is preferred that the additional-pass mask adjustment means is arranged to repeat the operation of the first-pass adjustment means, with the lower threshold, a number of times. For example, depending on the particular threshold levels used during the first and subsequent mask adjustment processes, the multiple-pass mask adjustment means may repeat the operation of the first-pass mask adjustment means (with the lower threshold level) at least twice or at least three times.

The mask adjustment means may be arranged to determine the said number of entries which indicate that the said association has been established in a group of nine entries corresponding to a 3x3 group of blocks in the input image. In this case, for example, the said predetermined threshold may be set to four or five for the first-pass mask adjustment process and to two or three for the subsequent mask adjustment processes.

The present invention also provides a method of motion compensated video signal processing in which motion vectors are generated to represent image motion between a pair of images of an input video signal, the method comprising:

deriving local motion vectors representing the motion of the content of respective blocks of one image of the pair between that image and the other image of the pair;

deriving from the local motion vectors a set of global motion vectors comprising a plurality of unique motion vectors selected from the most common of the local motion vectors;

establishing an association between each global vector and blocks of the said image by comparing each global motion vector with each local motion vector, the said association being established between a global motion vector and a given block if the global motion vector is within a predetermined tolerance of the local motion vector for that block; and

assigning to each said block a plurality of motion vectors selected from the zero motion vector, the local motion vector for that block, the local motion vectors for the blocks neighbouring that block in the input image, and any global motion vectors for which the said association with that block has been established.

The above method may be carried out using apparatus as hereinbefore described. In general, it is to be understood that, where features are described herein with reference to an apparatus according to the invention, corresponding features may be provided in accordance with a method of the invention, and vice versa.

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings in which:

Figure 1 is a schematic diagram of a video image showing artifacts due to the unrestricted application of global motion vectors;

Figure 2 is a schematic block diagram of motion compensated television standards conversion apparatus;

Figure 3 shows an example of an image having features which can give rise to errors in the conversion processing;

Figures 4, 6a and 6b are schematic cross-sections through respective correlation surfaces;

Figure 5 is a schematic representation of a correlation surface;

Figure 7 illustrates schematically a correlation surface having a ridge minimum.

Figure 8 illustrates how image artifacts can result from the presence of a ridge minimum in a correlation surface as shown in Figure 7;

Figure 9 is a schematic block diagram of part of a motion vector estimator of the standards conversion apparatus of Figure 2 for detecting ridge minima as shown in Figure 7;

Figures 10a, 10b and 10c illustrate the operation of the apparatus of Figure 9;

Figure 11 illustrates a further method of ridge minimum detection;

Figures 12a, 12b and 12c illustrate three classifications of correlation surface minima;

Figure 13 is a schematic block diagram of a motion vector reducer of the standards conversion apparatus of Figure 2;

Figure 14 illustrates the operation of a ridge minima requalifier of the motion vector reducer of Figure 13;

Figure 15 is a schematic block diagram of a ridge minima requalifier;

Figure 16 is a schematic block diagram of an alias vector validator of the motion vector reducer of Figure 13;

Figure 17 is a schematic block diagram of a global vector detector of the motion vector reducer of Figure 13;

Figure 18 is a schematic block diagram of a global vector restricter of the motion vector reducer of Figure 13;

Figure 19 is a diagrammatic representation of a global mask array for the image of Figure 1;

Figure 20 is a diagram used in explaining the operation of mask adjustment means of the global vector restricter of Figure 18;

Figure 21 shows the global mask array of Figure 19 after adjustment;

Figure 22 is a schematic block diagram of a block vector allocator of the motion vector reducer of Figure 13; and

Figure 23 is a diagram used in explaining the operation of a motion vector selector of the conversion apparatus of Figure 2.

Figure 2 is a schematic block diagram of motion compensated television standards conversion apparatus. The apparatus receives an input interlaced digital video signal 1 (eg a 1125/60 2:1 high definition video signal (HDVS)) and generates an output interlaced digital video signal 2 (eg a 1250/50 2:1 HDVS).

The input video signal 1 is first supplied to an input buffer/packer 3. In the case of a conventional definition input signal, the input buffer/packer 3 formats the image data into a high definition (16:9 aspect ratio) format, padding with black pixels where necessary. For a HDVS input the input buffer/packer 3 merely provides buffering of the data.

The data are passed from the input buffer/packer 3 to a matrix circuit 4 in which (if necessary) the input video signal's format is converted to the standard "CCIR recommendation 601" (Y, Cr, Cb) format.

From the matrix circuit 4 the input video signal is passed to a time base changer and delay 5, and via a sub-sampler 6 to a sub-sampled time base changer and delay 7. The time base changer and delay 5 determines the temporal position of each field of the output video signal, and selects the two fields of the input video signal which are temporally closest to that output field for use in interpolating that output field. For each field of the output video signal, the two input fields selected by the time base changer are appropriately delayed before being supplied to an interpolator 8 in which that output field is interpolated. A control signal t, indicating the temporal position of each output field with respect to the two selected input fields, is supplied from the time base changer and delay 5 to the interpolator 8.

The sub-sampled time base changer and delay 7 operates in a similar manner, but using spatially sub-sampled video supplied by the sub-sampler 6. Pairs of input fields are selected by the sub-sampled time base changer and delay 7 from the sub-sampled video, to be used in the generation of motion vectors.

The time base changers 5 and 7 can operate according to synchronisation signals associated with the input video signal, the output video signal, or both. In the case in which only one synchronisation signal is supplied, the timing of fields of the other of the two video signals is generated deterministically within the time base changers 5, 7.

The pairs of fields of the sub-sampled input video signal selected by the sub-sampled time base changer and delay 7 are supplied to a motion processor generally indicated at 9. The motion processor 9 comprises a direct block matcher 10, a data stripper 11, a motion vector estimator 12, a motion vector reducer 13, a motion vector selector 14 and a motion vector post-processor 15. The pairs of input fields are supplied first to the direct block matcher 10 which calculates correlation surfaces representing the spatial correlation between search blocks in the temporally earlier of the two selected input fields and (larger) search areas in the temporally later of the two input fields. Data representing these correlation surfaces are reformatted by the stripper 11 and are passed to the motion vector estimator 12. The motion vector estimator 12 detects points of greatest correlation in the correlation surfaces. (The correlation surfaces actually represent the difference between blocks of the two input fields. This means that the points of maximum correlation are in fact minima on the correlation surface, and are referred to as such herein.) In order to detect a minimum, additional points on the correlation surfaces are interpolated, providing a degree of compensation for the loss of resolution caused by the use of sub-sampled video to generate the surfaces. From the detected minimum on each correlation surface, the motion vector estimator 12 generates a local motion vector. The local motion vectors, one for each block of the first of the pair of input fields, are supplied to the motion vector reducer 13.

The motion vector estimator 12 also performs a confidence test on each generated local motion vector to establish whether that motion vector is significant above the general noise level, and associates a confidence flag with each motion vector indicative of the result of the confidence test. The confidence test, known as the "threshold" test, is described (along with other features of the apparatus of Figure 2) in GB-A-2 231 749.

Two further tests, an "alias" test and a "ridge" test, may also be performed in the motion vector estimator 12. These tests will be described in more detail below. Briefly, however, the alias test is performed to detect whether a local motion vector is potentially aliased. In this test, the correlation surface (apart from an exclusion zone around the detected minimum) is examined to detect the next lowest minimum. If this second minimum does not lie at the edge of the exclusion zone, the local motion vector derived from the original minimum is flagged as being potentially aliased. In the ridge test, a correlation surface is tested to check whether the detected minimum is a ridge minimum rather than a point minimum. The original minimum detected by the motion vector estimator is expanded to form a minimum region. This is performed by a thresholding process, starting from the originally detected minimum and including in the region any adjacent points within a preset range of the minimum value. The shape of the minimum region generated in this way depends on whether a point minimum or a ridge minimum is present. A local motion vector derived from the detected minimum is flagged to indicate detection of a ridge minimum.

The operation of the motion vector reducer 13 will also be described in detail below. Briefly, however, local motion vectors flagged as being potentially aliased, or corresponding to ridge minima, may be corrected in the motion vector reducer. The motion vector reducer 13 derives a set of global motion vectors, and assigns to each block of the first of the pair of input images a plurality of motion vectors selected, in order of preference, from the zero motion vector, the local motion vector for that block, the local motion vectors for the blocks neighbouring that block in the input image, and the global motion vectors. The assignment of a given global motion vector is restricted to parts of the input image at or near areas where motion identical or similar to that represented by the global motion vector has been detected.

The vectors assigned to the blocks by the motion vector reducer 13 are supplied to the motion vector selector 14. The motion vector selector 14 also receives as inputs the two input fields which were selected by the sub-sampled time base changer and delay 7 and which were used to calculate the motion vectors. These fields are suitably delayed so that they are supplied to the motion vector selector 14 at the same time as the vectors derived from those fields. The motion vector selector 14 derives an output comprising one motion vector per pixel of the output field. The motion vectors corresponding to a given block of the input field are tested for pixels within a corresponding block of the output field to be generated. For each output pixel, a motion vector is selected from the motion vectors supplied for the corresponding block of the input field.

The vector selection process involves detecting the degree of correlation between test blocks of the two input fields pointed to by a motion vector under test. The motion vector having the greatest degree of correlation between the test blocks is selected for use in interpolation of the output pixel. A "motion flag" is also generated by the vector selector. This flag is set to "static" (no motion) if the degree of correlation between blocks pointed to by the zero motion vector is greater than a preset threshold.

The vector post-processor 15 reformats the motion vectors selected by the motion vector selector 14 to reflect any vertical scaling of the picture, and supplies the reformatted vectors to the interpolator 8. Using the motion vectors, the interpolator 8 interpolates an output field from the corresponding two (non-sub-sampled) interlaced input fields selected by the time base changer and delay 5, taking into account any image motion indicated by the motion vectors currently supplied to the interpolator 8.

If the motion flag indicates that the current output pixel lies in a moving part of the image, pixels from the two selected fields supplied to the interpolator are combined in relative proportions depending on the temporal position of the output field with respect to the two input fields (as indicated by the control signal t), so that a larger proportion of the nearer input field is used. If the motion flag is set to "static", then temporal weighting is not used. The output of the interpolator 8 is passed to an output buffer 16 for output as an HDVS output signal, and to a down-converter 17 which generates a conventional definition output signal 18, using the motion flag.

The sub-sampler 6 performs horizontal and vertical spatial sub-sampling of the input fields received from the matrix 4. Horizontal sub-sampling is a straightforward operation in that the input fields are first pre-filtered by a half-bandwidth low pass filter (in the present case of 2:1 horizontal decimation) and alternate video samples along each video line are then discarded, thereby reducing by one half the number of samples along each video line.

Vertical sub-sampling of the input fields is complicated by the fact that the input video signal is interlaced. This means that successive lines of video samples in each interlaced field are effectively separated by two video lines of the complete frame, and that the lines in each field are vertically displaced from those in the preceding or following field by one video line of the complete frame. The method of vertical sub-sampling actually used involves a first stage of low pass filtering in the vertical direction (to avoid aliasing), followed by a filtering operation which effectively displaces each pixel vertically by half a video line downwards (for even fields) or upwards (for odd fields). The resulting displaced fields are broadly equivalent to progressively scanned frames which have been sub-sampled vertically by a factor of two.

The alias test which may be performed by the motion vector estimator 12 will now be described in more detail. As explained earlier, the correlation surfaces are derived in the block matcher 10 by comparing a block (for this purpose called a "search block") of the first of the pair of input images with a (larger) search area in the second of the pair of input images. The correlation surface represents the difference between the search block and the search area for each position of the search block in the search area, and the local motion vector is obtained from the point of maximum correlation, ie the minimum, on the correlation surface.

A problem can occur with images having areas of regularly spaced patterning, such as the image illustrated in Figure 3. The image is subject to horizontal motion (resulting from a horizontal camera pan). A search block 20 used to generate a correlation surface encompasses more than one "cycle" of the regularly spaced patterning. When motion vector estimation is performed on the image shown in Figure 3, multiple minima can be detected which can be spaced across the correlation surface. The multiple minima are illustrated in Figure 4 which is a schematic representation of a cross-section through a correlation surface generated from the search block 20. The minimum 21 represents the correct motion of the scene, and the other minimum 22 results from alias interactions between the image patterning and the size of the blocks used to generate the correlation surfaces. The vector estimation process has no way of identifying which one of the minima represents the correct image motion, so that an aliased (incorrect) vector may be generated. If this aliased vector is subsequently used in interpolation of an output image, subjectively disturbing image artifacts can result. The principle of the alias test which is used to detect when a motion vector derived from a correlation surface is potentially aliased can be understood by considering Figures 5, 6a and 6b.

Figure 5 is a schematic diagram of a correlation surface 25 representing the difference between a search block and a search area in the first and second of the pair of input fields respectively. The point of maximum correlation is represented by the minimum 26 in the correlation surface, and the position of the minimum 26 determines the magnitude and direction of the local motion vector obtained from the correlation surface 25.

In order to detect whether the motion vector derived from the correlation surface 25 is potentially aliased, an exclusion region 27 (for example a 3x3 array of correlation values around the minimum 26) is defined around the minimum 26. The correlation surface (apart from the exclusion region 27 around the minimum 26) is then re-examined to identify the next lowest minimum. If the next minimum does not lie at the edge of the exclusion region then the generated motion vector is flagged as being potentially aliased.

Figures 6a and 6b are schematic representations of cross sections through two correlation surfaces 28 and 29. The correlation surface 28 in Figure 6a has a single minimum 30 and therefore does not suffer from alias problems as described above. When an exclusion region 31 is defined around the minimum 30, the next lowest minimum 32 is found to lie at the edge of the exclusion region. This indicates that the surface is not aliased and an alias flag associated with the motion vector generated from the correlation surface 28 is set to indicate a "not aliased" condition.

In contrast, the correlation surface 29 shown in Figure 6b has a number of distinct minima 34, 35, 36. This correlation surface represents the correlation between two images having regularly spaced patterning. During motion vector estimation from the surface 29, the lowest minimum 34 (the point of maximum correlation) is detected and an exclusion region 37 is defined around the minimum 34. The next lowest minimum is then found to be the minimum 35 which does not lie at the edge of the exclusion region. The motion vector generated from the minimum 34 is therefore flagged as being potentially aliased, an alias flag associated with the vector being set to indicate an "aliased" condition.

Local motion vectors flagged as potentially aliased as a result of the alias test are re-examined by an alias vector validator during the motion vector reduction process in the motion vector reducer 13 which will be described in detail below.

The ridge test which may be performed in the motion vector estimator 12 will now be described in more detail with reference to Figures 7 to 12.

The purpose of the ridge test is to identify local motion vectors associated with ridge minima in correlation surfaces. A ridge minimum can occur if the image contains moving objects (eg lampposts) which, in one direction, are longer than the size of the search areas used in block matching. In this case, because the moving object extends out of the search area used in block matching, it is not possible (within that search area) to detect the component of the object's motion along the object's longitudinal axis. The result is that a "one-dimensional" or ridge minimum (a ridge correlation maximum) is detected.

Figure 7 illustrates an example of a correlation surface 40 having a ridge minimum 41. The ridge minimum defines a component of the object's motion in a direction perpendicular to the ridge direction (the ridge direction lying along the longitudinal axis of the object), but gives no indication of the motion of the object along the ridge direction.

In previously proposed motion vector estimators, motion vectors generated from ridge minima have failed the confidence (or "threshold") test referred to earlier and described in GB-A-2 231 749, and such vectors have therefore not been used in interpolation of the output image. Instead, the zero motion vector has been used by default. However, using the zero motion vector to interpolate pixels in a long moving object can result in artifacts such as the omission of portions of the object. An artifact of this type is illustrated in Figure 8 which shows the interpolation of an output image having a long vertically moving object 45. A correct motion vector 46 is detected at the top and bottom of the moving object, but a default zero motion vector 47 is selected for the central portion of the object. A gap 48 is formed between the upper portion interpolated using the correct motion vector and the central portion interpolated using the zero motion vector.

Figure 9 is a block diagram of apparatus for identifying ridge minima during motion vector estimation in the motion vector estimator 12. As shown in Figure 9, correlation surfaces comprising an array of correlation values are received from the data stripper 11 and are stored in a correlation surface store 50. The correlation values are read from the correlation surface store 50 by a "first" minimum detector 51, which scans the correlation surface to detect a single minimum value from the correlation surface. In fact the first minimum detector 51 outputs three items: the actual minimum correlation value (MV) 52; the average correlation value (AV) 54 of the correlation surface; and the coordinates (or position) (MC) 53 of the minimum value 52 in the correlation surface.

After this first scan has been performed to detect the minimum value (MV) 52, a second examination of the correlation surface is performed by a "next" minimum detector 55. The next minimum detector 55 receives the correlation values from the correlation store 50, along with the coordinates of those correlation values. The coordinates of the correlation values are also supplied to an exclusion generator 56 which generates an exclusion flag 57 to mask an exclusion region of, for example, 5x5 correlation values around the coordinates 53 of the minimum value 52. The next minimum detector 55 then detects the minimum value outside of the exclusion region. This detected next minimum value is subjected to a threshold comparison in which the difference between the detected next minimum value and the minimum value 52 is compared with a threshold value derived using the average value 54 of the correlation surface. If the difference between the minimum value 52 and the next minimum value is greater than the threshold value, a "valid vector" flag (VV) 58 is set to indicate that the first minimum value 52 represents a significant correlation peak.

A third examination of the correlation surface is performed by a "similar" minimum detector 60. The similar minimum detector detects all correlation values in the correlation surface lying within a threshold value of the minimum value 52. Again, the threshold value used in the similar minimum detector 60 is derived using the average value 54 of the correlation surface. As each of these points is detected by the similar minimum detector 60, a count held by a minima counter and comparator 61 is incremented.

The similar minimum detector 60 detects the position of the two correlation values within the threshold of the minimum value 52 which are most widely spaced in the correlation surface. The coordinates 62 of these most widely spaced minima are passed to a vector averager 63.

The vector averager 63 generates an output 64 representing a motion vector derived from the mean position of the most widely spaced minima 62 as detected by the similar minimum detector 60. This output 64 is passed to an output vector selector 65. The vector averager 63 also detects the spatial relationship of the most widely spaced minima 62, to detect whether the ridge lies in a substantially horizontal or a substantially vertical direction. The result of this test is output as a "ridge direction" flag 66.

The valid vector flag 58 and the minimum coordinates (MC) 53 are supplied to a vector generator 67, in which (if the first minimum is valid) a motion vector is generated from the minimum coordinates 53. This motion vector is passed to the output vector selector 65.

The count held by the minima counter 61 is compared with two threshold values indicative of a ridge range 68. If the count held by the minima counter 61 indicates that more than a first predetermined number, but fewer than a second predetermined number, of minima have been detected in the correlation surface, a "valid ridge" flag (VR) 69 and a "ridge" flag 70 are set to indicate that a ridge minimum has been detected but valid use can be made of the ridge vector 64 generated by the vector averager 63. In this case, the output vector selector supplies the ridge vector 64 as an output vector 71.

If more than the second predetermined number of minima have been detected, the valid ridge flag 69 is not set, but the ridge flag 70 is set. This indicates that the ridge vector 64 should not be used, and the motion vector should either be re-qualified (see below) or not used. In this case, the output vector selector supplies the motion vector generated by the vector generator 67 as the output vector 71.

If fewer than the first predetermined number of minima have been detected, neither the valid ridge flag 69 nor the ridge flag 70 are set. Again, in this case, the output vector selector supplies the motion vector generated by the vector generator 67 as the output vector 71.

Figures 10a, 10b and 10c illustrate the operation of the apparatus of Figure 9 when the valid ridge flag 69 is set.

Figures 10a and 10b illustrate two successive images having an elongate vertical object 75 with a horizontal motion component 76 of $\underline{v}$. When blocks 77, 78 of the images shown in Figures 10a and 10b are compared to generate a correlation surface 79 (Figure 10c), a large number of minima 80 are generated having very similar correlation values.

When the correlation surface 79 is processed by the apparatus of Figure 9, the two most widely spaced minima 81, 82 are passed to the vector averager 63. The vector averager 63 generates a ridge vector 64 equal to ($\underline{va}$+$\underline{vb}$)/2, ie a motion vector representing the mean position of the two minima 81, 82. Assuming that the number of minima 80 exceeds the predetermined number, the ridge vector 64 is output by the vector output selector 65 as the output vector 71.

Figure 11 illustrates a second, similar, method of ridge minimum detection. In a first examination of a correlation surface 90, the minimum correlation value 91 is detected. All correlation values within a threshold of the minimum value 91 are then detected, to define a threshold region 92. However, in contrast to the method described with reference to Figure 9, the number of minima within the threshold of the minimum value 91 is not detected. Instead, an 'aspect ratio', ie the ratio of length (in the vertical direction) to width (in the horizontal direction) of the threshold region 92 is detected. From the aspect ratio, the threshold region can be classified into one of three classifications illustrated in Figures 12a, 12b and 12c.

Referring to Figure 12a, if the detected aspect ratio lies between a first and a second predetermined value, indicating that the threshold region 93 is not elongate, the minimum is classified as a point minimum having defined coordinates in both the horizontal (x) and vertical (y) directions.

In Figure 12b, a minimum region 94 has an aspect ratio of greater than the first predetermined value, indicating that the region is elongate in the vertical direction 95. The minimum is then classified as a vertical ridge minimum and the resulting motion vector has a defined component 96 in the horizontal direction but an undefined component in the vertical direction 95.

Similarly, Figure 12c illustrates a threshold region 97 having an aspect ratio of less than the second predetermined value, ie indicative of a horizontal ridge minimum. The resulting motion vector has a defined vertical component but an undefined horizontal component.

From the classification of the threshold regions described above, a ridge flag is associated with each motion vector to indicate the classification of minimum from which that motion vector was derived. In other words, the ridge flag (for which two bits will suffice) indicates whether the motion vector was derived from a point minimum, a vertical ridge minimum or a horizontal ridge minimum.

It will be appreciated that, where the alias test referred to earlier is carried out during motion vector estimation in addition to the ridge test just described, then the alias flags may be derived at the same time as the ridge test is performed. For example, in the apparatus of Figure 9, the alias flags could be derived by an exclusion generator and "next minimum" detector similar to the components 56 and 55 described with reference to Figure 9. Similarly, the confidence test referred to earlier (and described in GB-A-2 231 749) may be carried out in parallel with the alias test and/or the ridge test.

When local motion vectors are derived from ridge minima as described above, the motion vectors may be "repaired" by a ridge minima requalifier in the motion vector reducer 13. The construction and operation of the motion vector reducer 13 will now be described in more detail.

A block diagram of the motion vector reducer 13 is shown in Figure 13. The motion vectors, and their associated flags, derived by the motion vector estimator 12 are supplied via an input 100 of the motion vector reducer 13 to a ridge minima requalifier 101. The ridge minima requalifier 101 "repairs" motion vectors derived from ridge minima and supplies the vectors, and associated flags, to an alias vector validator 102 and a global vector detector 103. The alias vector validator 102 performs a test to check whether vectors indicated as potentially aliased can be justifiably requalified as non-aliased, and the resulting vectors, and associated flags, form one input to a global vector restricter 104. The global vector detector 103 derives a set of global motion vectors from the vectors supplied thereto. The global vectors output by the global vector detector 103 form a second input to the global vector restricter 104.

The global vector restricter 104 generates, for each global vector supplied thereto, a global mask array indicating an association between the global vector and various blocks of the input image to which the local vectors supplied from the alias vector validator 102 correspond. The global mask arrays generated by the global vector restricter 104 are supplied to a block vector allocator 105 which also receives the (suitably delayed) local vectors, and associated flags, from the alias vector validator 102 and the global vectors from the global vector detector 103. The block vector allocator 105 assigns to each block of the input image to which the local vectors correspond a plurality of unique vectors which are then supplied via an output 107 of the motion vector reducer 13 to the motion vector selector 14.

The ridge minima requalifier 101, alias vector validator 102, global vector detector 103, global vector restricter 104 and block vector allocator 105 operate under the control (indicated schematically by dotted lines in the figure) of a microprocessor 106. The microprocessor 106 is also connected to the global vector detector 103 to provide for optional manual intervention in the global vector detection process to adjust various parameters. The adjusted parameters can be entered by an operator and supplied by the microprocessor 106 to the global vector detector 103. A two-way communication link 108 is also provided to connect the microprocessor 106 to a further, similar microprocessor (not shown). This is because the entire motion processor (indicated generally at 9 in Figure 2) may in fact be duplicated, the two motion processors performing the processing for alternate output fields. Thus, the vector reducer shown in Figure 13 may be duplicated, and the link 108 is provided for communication of the two microprocessors controlling the vector reduction operation.

The construction and operation of the various components of the motion vector reducer 13 will now be described in more detail.

Considering first the ridge minima requalifier 101, this receives the local vectors, and associated flags, from the motion vector estimator 12 via the input 100 to the motion vector reducer 13. As described earlier, motion vectors derived from ridge minima, and identified as such during motion vector estimation, will have a ridge flag associated therewith which identifies the nature of the ridge. For the system previously described with reference to Figures 11 and 12, the ridge flag is a 2-bit flag which indicates whether a given motion vector is derived from a point minimum, a vertical ridge minimum, or a horizontal ridge minimum. When a motion vector has been derived from a ridge minimum, the component of the motion vector in the ridge direction (eg the vertical direction in the case of a vertical ridge minimum) is undefined. The ridge minima requalifier 101 attempts to requalify motion vectors derived from horizontal or vertical ridge minima by assigning a valid vector component in the ridge direction. This requalification of motion vectors in the ridge minima requalifier 101 is illustrated schematically in Figure 14.

In the ridge vector qualifier 101, motion vectors are stored, with their associated flags, in an array memory 120 at positions corresponding to the relative positions of the search blocks from which the motion vectors were derived. During an examination of the motion vectors, if a motion vector derived from a ridge minimum is encountered, adjacent motion vectors **in the ridge direction** are examined to detect whether the vector component at 90° to the ridge direction matches (i.e lies within a threshold of) the corresponding component of the motion vector derived from the ridge minimum.

If a match is found, it is assumed that the adjacent vector relates to the same elongate moving object. If the adjacent vector was itself derived from a ridge minimum, the search is continued **in the ridge direction**. Eventually, a motion vector may be found which has a substantially identical component at 90° to the ridge direction but was derived from a point minimum (ie it has a defined component along the ridge direction). It is assumed that this motion vector represents the motion of the end of the elongate object, and therefore that the component of this motion vector in the ridge direction is equal to the previously undefined component in the ridge direction of the motion vector under examination.

Two examples of this process (one horizontal, one vertical) are illustrated schematically in Figure 14. In the vertical example, a motion vector 121 has a ridge flag indicating that it was generated from a vertical ridge minimum, and accordingly has a good horizontal (x) component but an undefined vertical (y) component. An adjacent motion vector 122 in the ridge direction is examined and is found to have a matching x component but an undefined y component (the vector 122 also being generated from a vertical ridge minimum). Similarly, further searches of vectors 123 and 124 also indicate a matching x component but an undefined y component. Eventually, a motion vector 125 is detected having a matching x component and a defined y component (this motion vector being generated from a point minimum). It is assumed that the motion vector 125 represents the motion of the end of the elongate object, and accordingly the y component of the motion vector 125 is assigned to the motion vector 121. The ridge flag associated with the motion vector 121 is modified to indicate that the motion vector 121 has been re-qualified (by setting the ridge flag to indicate that the motion vector 121 was derived from a point minimum), and the confidence flag associated with the vector is reset, so that the requalified vector can subsequently be used in interpolation of an output pixel.

A corresponding process is illustrated for the re-qualification of a motion vector 130 derived from a horizontal ridge minimum. In this case, a first search in a direction from left to right is unsuccessful, because the edge of the array 120 of the motion vectors is reached. A second search is then carried out, in a direction from right to left, in which three motion vectors 131 are detected which have y components matching the motion vector 130 but undefined x components. Eventually, a motion vector 132 is detected which has a y component matching that of the motion vector 130, and a defined x component. The x component of the motion vector 132 is assigned to the motion vector 130 and the ridge and confidence flags modified as described above.

Figure 15 is a schematic block diagram of a ridge vector requalifier 101 for performing the above process. Motion vectors received from the motion vector estimator 12 are stored in an input vector store 140 along with their associated flags. A vector processor 141 reads the stored motion vectors and performs the re-qualification process described above with reference to Figure 14. The possibly re-qualified motion vectors and flags output by the vector processor 141 are stored in an output vector store 142, before being passed for further vector reduction processing. The use of separate input and output vector stores 140, 142 avoids the propagation of errors through the array of vectors as a result of the re-qualification process.

The vectors, and their associated flags, output by the ridge minima requalifier 101 are supplied to the alias vector validator 102. The alias vector validator 102 attempts to requalify motion vectors which have been flagged as being potentially aliased during the motion vector estimation process as described earlier. A schematic block diagram of the alias vector validator 102 is shown in Figure 16.

Each of the motion vectors supplied to the alias vector validator 102 is stored at a position in a memory array 150 depending on the position (in the input image) of the search block from which the corresponding correlation surface was derived. The ridge and confidence flags associated with the motion vectors are stored in a corresponding confidence flag array 151 and ridge flag array 152, and the alias flags (generated as described above) are stored in an input alias flag array 153.

The re-qualification of potentially aliased motion vectors comprises a sequential scan through the array memories 150, 151 and 153 under the control of an address generator 155. For each of the motion vectors, the alias flag, the vector value (magnitude and direction) and the confidence flag for that motion vector 156 are supplied to a comparator 157. The address generator 155 also controls the reading of the alias flags, confidence flags and vector values for the 24 motion vectors surrounding the vector under test in a 5 x 5 array pattern 158. These 24 sets of flags and values are also supplied to the comparator 157.

The comparator 157 tests the vector under test against the surrounding 24 vectors according to a set of rules as given below, and generates a modified alias flag for the vector under test. This modified flag is written to a corresponding position 159 in an output alias flag array 160, under the control of the address generator 155.

The rules obeyed by the comparator 157 are as follows:

1. If the confidence flag for the vector under test is not set (indicating that the vector failed the confidence test) then the corresponding alias flag is written without modification into the output alias flag array 160. In this case, the state of the alias flag is immaterial, since a vector which failed the confidence test is not used in motion compensated interpolation.

2. If the confidence flag is set and the alias flag is not set for the vector under test (indicating that the vector under test passed the confidence test and is not aliased), then the alias flag for the vector under test is written without modification into a corresponding position in the output alias flag array 160.

3. If the alias flag and the confidence flag for the vector under test are set (indicating that the vector passed the confidence test but is potentially aliased), then:

a) if the value of the vector under test lies within a preset threshold 161 of the value of one or more of the surrounding 24 vectors (the one or more vectors being non-aliased and having passed the confidence test) then the alias flag for the vector under test is re-qualified to indicate "not-aliased". The re-qualified alias flag is written into the output alias flag array 160;

b) if the value of the vector under test does not lie within the preset threshold 161 of the value of one or more of the surrounding 24 vectors, then the alias flag for the vector under test is not re-qualified, and is written without modification into the output alias flag array 160.

The use of separate input and output alias flag arrays prevents the propagation of errors through the array during aliased vector requalification.

After the above analysis has been performed by the alias vector validator 102, the vectors, and associated flags, are read out by the address generator 155 for output to the next stage of the vector reduction process.

The vectors, and flags, output by the ridge minima requalifier 101 are also supplied to the global vector detector 103. The purpose of the global vector detector is to derive from the vectors supplied thereto a set of global motion vectors comprising a plurality of unique motion vectors selected from the most common of the local motion vectors. In general terms, the global vector detector 103 ranks local motion vectors corresponding to a given input image in order of decreasing frequency, and selects the eight (in this example) most common vectors as global vectors. However, it can be advantageous to apply various constraints to the process for selecting global vectors from the set of local vectors. For example, even though large magnitude (long) motion vectors (ie vectors representing rapid image motion) can be generated, it is desirable to prevent such long vectors from contributing to the global vectors. This is because the lack of spatial detail in fast-moving objects (eg due to camera integration blur) means that the operation of the motion vector selector 14 (Figure 2) for rapidly moving objects can be unreliable. It has been observed that rapidly moving portions of a video image have lower spatial detail than stationary or slowly moving portions. This means that even if a motion vector can be estimated which accurately represents the rapid motion of the object, the correlation test performed in the vector selector 14 (see below) which relies on detecting correlation between parts of the same object in two successive input fields can be unreliable. It is therefore preferable to prevent such long vectors from being selected as global vectors to ensure that a large magnitude global vector cannot be incorrectly selected for interpolation of image areas where there is in fact only very small motion. This could produce subjectively disturbing effects in the output image.

A further constraint which may be applied to the global vector detection process is to prevent a vector being accepted as a global vector unless it occurs at least a predetermined number of times over the input image as a whole. If a local motion vector is not sufficiently common, and thus represents motion of only a very small part of the image, it may be preferable to eliminate the possibility of this vector being assigned to other parts of the image during the global vector detection stage rather than relying solely on the global vector restriction technique which will be described later.

A schematic block diagram of a global vector detector 103 in which the above constraints are implemented is shown in Figure 17. The input vectors, and associated flags, are supplied to an amplitude detector 170 which compares the amplitude, ie magnitude, of each input vector with a predetermined magnitude threshold 171 to decide whether the vector qualifies as a "long" vector. The magnitude threshold 171 may comprise a threshold for both the horizontal and vertical components of a motion vector. For example, the threshold may be set at half the maximum possible range, horizontally and vertically, of a motion vector. In this example, the maximum horizontal range of a motion vector is ±64 pixels, and the maximum vertical range ±32 pixels. Thus, the magnitude threshold 171 may be set at ±32 pixels horizontally and ±16 pixels vertically. If the magnitude of an input vector exceeds the magnitude threshold 171, the amplitude detector 170 sets a "long vector" flag (LV).

The input vectors supplied to the amplitude detector 170 are also supplied to a frequency array 172. Each entry in the frequency array

172 is uniquely addressed by a possible value of the local motion vectors. In this way, a local motion vector generated from the current pair of input fields is used to address the array and to increment the array entry corresponding to the value of the motion vector. However, the long vector flags LV derived by the amplitude detector 170 are also supplied to the frequency array 172. If the long vector flag LV for a motion vector supplied to the frequency array 172 is set, then this prevents the associated motion vector from incrementing the frequency array.

After all the vectors (apart from long vectors) have been counted in this way by the frequency array 172, an array scanner 173 scans the frequency array 172 and identifies the eight array entries which show the highest count. The addresses of these array entries represent the values of the eight most common local vectors (apart from long vectors) for the corresponding pair of input images. The value of the count held by the frequency array for each of these eight vectors (that is, the frequency of occurrence of the vectors) is supplied by the array scanner 173 to a comparator 174 at the same time as the corresponding vector is supplied to an input of a switch 175. The comparator 174 compares the count, or frequency of occurrence, corresponding to each vector with a predetermined threshold number or "global threshold", and provided the count exceeds the global threshold, the corresponding vector will be supplied to the output 176 of the switch 175 as a global motion vector. (A suitable value for the global threshold may vary depending on the particular image material being processed. By way of example, however, the global threshold may be set at 0.5% of the total number of local motion vectors derived from an input image.) If the count corresponding to any of the eight most common vectors does not exceed the global threshold, indicating that the vector does not occur sufficiently frequently to qualify as a global vector, then the comparator 174 generates an output to control the switch 175 such that the corresponding vector is not passed to its output 176 as a global vector. Assuming, however, that all eight of the most common vectors pass the global threshold test, then a set of eight unique global motion vectors will be supplied to the output 176 of the switch 175.

It will be appreciated that various further constraints could be applied in the global vector detector 103. In particular, any input local motion vectors for which the corresponding ridge or alias flags are set, or for which the confidence flag indicates that the vector failed the confidence test, may be prevented from contributing to the global vectors in a similar manner to long vectors, ie by preventing such vectors from incrementing the frequency array 172. In addition, a constraint may be applied in the array scanner 173 to cause the array scanner 173 to identify the eight most common vectors which differ from each other by a predetermined minimum amount.

As indicated in Figure 17, the long vector flag LV derived for each of the input local motion vectors is supplied by the global vector detector 103, after an appropriate delay, to the block vector allocator 105 (Figure 13) for use in the subsequent vector reduction processing of the local vectors. This will be described in detail below.

As shown in Figure 13, the global vector detector 103 is connected for communication with the microprocessor 106. This allows for the various thresholds used during global vector detection to be modified if required through manual intervention by an operator who can enter new parameters, for example via a system controller, which are then supplied via the microprocessor 106 to the global vector detector 103.

The global vectors output by the global vector detector 103 are supplied to one input of the global vector restricter 104. At its other input, the global vector restricter 104 receives the processed local motion vectors and associated flags from the alias vector validator 102. The purpose of the global vector restricter 104 is to associate each of the global vectors with blocks of the input image at or near areas where motion identical or similar to that represented by the global motion vector was originally derived. This association will then be used by the block vector allocator 105 to prevent inappropriate global vectors being assigned to blocks, that is, to prevent a block from one area of the input image being assigned a global vector representing significantly different motion of a block in a remote part of the image. A schematic block diagram of the global vector restricter 104 is shown in Figure 18.

Referring to Figure 18, the local motion vectors, and associated flags, output by the alias vector validator 102 are stored in a vector array memory 180. Each local vector is stored at a position in the array memory 180 corresponding to the position of the associated search block in the first of the pair of input images from which the local vectors were derived. Each of the eight global vectors derived by the global vector detector 103 is supplied to one input of a respective one of eight comparators indicated schematically at 181. (In fact, each of the eight global vectors may be stored in a respective global vector store, the content of which is read repeatedly and supplied to an input of the corresponding comparator 181 as required.)

Under control of an address generator 182, the local motion vectors are read in turn from successive locations in the vector array 180, and as each local vector is read out, it is supplied to a second input 183 of each of the eight comparators 181. The comparators 181 then simultaneously compare the local motion vector with each of the eight global motion vectors supplied to the first inputs of the comparators.

The output of each of the comparators 181 controls a corresponding mask array memory 184 which is also addressed by the address generator 182. In each comparator 181, if the global motion vector is within a predetermined tolerance 185 of the local motion vector currently supplied to the comparator, then the comparator 181 sets a mask bit to "1" at a position in the corresponding mask array memory 184 corresponding to the position of the local motion vector in the vector array 180. If the global vector is not within the predetermined tolerance 185 of the local motion vector, then the comparator 181 sets the mask bit to "0". A typical value for the tolerance 185 supplied to the comparators 181 may be, for example, ±1 pixel horizontally and vertically.

Thus, as each of the local motion vectors is read out in turn from the vector array 180, a mask bit is set in each of the mask array memories 184 to indicate whether or not that local motion vector is identical or similar to the global motion vector to which each mask array memory 184 corresponds. The position of the mask bit set as a result of comparison of a given local motion vector corresponds to the position of that vector in the vector array 180, and thus to the position in the original input image of the block to which the local vector corresponds. As this is repeated for all the local vectors for the input image, a global mask array, or bit field, is built up in each array memory 184, each global mask array indicating an association, or correspondence, between the global motion vector and blocks of the original input image. A schematic representation of a global mask array for a global motion vector is shown in Figure 19.

In Figure 19, the global mask array 190 corresponds to an original input image as shown in Figure 1. The global mask array 190 illustrated is that obtained through comparison of a global motion vector which corresponds to the motion of the vehicle in Figure 1 with the local motion vectors derived over the image as a whole. The black squares indicate "set" mask bits, ie mask bits, or array entries, which have been set to "1" indicating an association between the global motion vector and blocks located at corresponding positions in the input image. As expected, therefore, most of the mask bits corresponding to blocks containing part of the moving vehicle in Figure 1 have been set in the mask array 190. However, it will be noted that, due to errors introduced, for example, through noise in the input signal, certain mask bits corresponding to blocks within the body of the moving vehicle have not been set. Equally, the global mask array 190 shows that certain mask bits corresponding to stationary portions of the input image have been set. To eliminate such errors in the global masks, each of the global masks stored in the mask memories 184 in Figure 18 is subjected to a multiple-pass mask adjustment processing operation which is effectively a multiple-pass filtering operation performed by respective multiple-pass filters 186 each of which is connected to a corresponding mask array memory 184. The operation during the first pass of the filtering process will now be described with reference to Figure 20.

Figure 20 represents an enlarged portion of a global mask array such as the mask array 190 shown in Figure 19 in which the dark squares represent set mask bits (value "1") and the white squares represent unset mask bits (value "0"). During the first pass of the filtering operation by a filter 186, each mask bit is considered together with the eight mask bits of a 3x3 group of mask bits surrounding the mask bit under test (the centre mask bit). For each 3x3 group, the number of set mask bits is determined and compared with a predetermined first-pass threshold. (In this example, the first-pass threshold may be set to 4 or 5 for example.) If the number of set mask bits exceeds the first-pass threshold, and the centre mask bit is not set (as is the case in the illustration of Figure 20), then the centre mask bit is set. Thus, if a sufficient number of mask bits surrounding a given unset mask bit are set, then the unset mask bit is reset to "1" indicating a set mask bit. If the total number of set mask bits in the 3x3 group does not exceed the first-pass threshold, and the centre mask bit is not set, then the centre mask bit is unchanged. Conversely, if the total number of set mask bits in the 3x3 group does not exceed the first-pass threshold, but the centre mask bit is set, then the centre mask bit is reset to "0" indicating an unset mask bit. If the total number of set mask bits in the 3x3 group exceeds the first-pass threshold and the centre mask bit is set, then the centre mask bit is unchanged. The resetting of mask bits during the filtering process does not affect the remainder of that pass of the filtering operation. For example, the filtered mask bits could be stored in a separate memory to the memory array from which the unfiltered mask bits are being read. Here, however, the filters are arranged to effect a sufficient delay between reading of unfiltered mask bits and output of filtered mask bits to prevent the fact that a mask bit has been reset from being taken into account during the remainder of that pass of the filtering operation.

It will be seen that, with a relatively high first-pass threshold of 4 or 5, the effect of the first-pass filtering operation on a global mask array such as that shown in Figure 19 would be to "fill in" spurious gaps in the region of the mask array 190 corresponding to the vehicle in Figure 1, and equally to eliminate the spurious set mask bits corresponding to stationary parts of the image of Figure 1. Thus, the effect of the first-pass filtering operation is to eliminate spurious errors in the original global mask array.

A side effect of the first-pass process, however, is that it "shrinks" the overall area of the group of set mask bits corresponding to the vehicle in Figure 1. This is because, for set mask bits corresponding to an outer edge of the vehicle, the mask bits for blocks beyond the edge will not be set, and, if there is a sufficient number of these, the originally set mask bits corresponding to the edge of the vehicle will be reset to "0". To counter this effect, the filtering operation is repeated with the threshold for the additional pass or passes of the filtering operation being reduced. For example, after the first pass, the global masks produced by the first-pass filtering operation may be reapplied to the filters 186 with the predetermined threshold set to 2 or 3. This has the effect of "spreading" the areas of set mask bits in the global mask arrays to cover approximately the same area as in the original unfiltered mask arrays (without, of course, the spurious errors). It is preferred that the filtering operation is repeated a number of times (at the lower threshold) after the first pass to produce a slight overall spreading in the areas of set mask bits in the global masks as compared with the original unfiltered masks. In the present example, it is preferred that the filtering operation is repeated three, or possibly four, times at the lower threshold after the first-pass filtering operation at the higher threshold. This has the effect of extending the application of a given global vector to blocks just beyond those neighbouring a block for which the local vector was within the tolerance 185 of the global vector. Vectors ("neighbouring vectors") corresponding to blocks neighbouring a given block may be assigned to that block in any case during the block vector allocation process (see below). Thus, sufficient additional passes of the filtering operation are performed to extend the application of the global vectors beyond blocks neighbouring those for which the corresponding mask bits were set in the unfiltered global masks.

For the unfiltered global mask array 190 shown in Figure 19, the final filtered global mask array 192 may be as shown in Figure 21 for example, where it can be seen that the set mask bits indicate far more accurately the association between the global motion vector for the vehicle of Figure 1 and the area of the vehicle within the image.

It will of course be appreciated that the filters 186 can be implemented as two-dimensional FIR (finite impulse response) 3x3 filters, with all coefficients set to unity, and the nine taps being connected to an adder the output of which forms an input to a comparator. The relevant threshold can be supplied to the other input of the comparator. The output of the comparator forms the filtered global mask array and can be reapplied to the filter input for the next pass with the threshold adjusted as necessary.

The filtered global mask arrays output by the filters 186, one for each global vector, are output by the global vector restricter 184 to the block vector allocator 105 in Figure 13. The block vector allocator 105 also receives the (appropriately delayed) vectors, and associated flags, from the alias vector validator 102 and the original eight global vectors from the global vector detector 103. The purpose of the block vector allocator 105 is to assign to each of the blocks of the original input image a set of four unique motion vectors including the zero motion vector. The motion vectors allocated to each block will be supplied to the motion vector selector 14 which will select from the four vectors for a given block the most appropriate motion vector to be assigned to each output pixel in a corresponding block of the output field. A schematic block diagram of an embodiment of the block vector allocator 105 is shown in Figure 22.

As shown in Figure 22, the local motion vectors, and flags, output by the alias vector validator 102 are supplied to an input 200 of the block vector allocator 105 and stored in a local vector array memory 201 at relative positions corresponding to the positions of the blocks in the original input image to which the vectors correspond. The local motion vectors, and flags, stored in the local vector array 201 are read out sequentially under the control of an address generator 205 and supplied to a flag analyser 203. Each of the eight global motion vectors (hereinafter designated by g1 to g8) are supplied to respective first inputs of a series of eight switches S1 to S8. (In fact, the global vectors g1 to g8 may be stored and repeatedly supplied to the respective switches S1 to S8 as required during the operation of the block vector allocator 105.) The eight global mask arrays (hereinafter gm1 to gm8), one for each global vector g1 to g8, are supplied to respective control inputs of the switches S1 to S8, each global mask array being supplied to the switch to which the corresponding global vector g1 to g8 is supplied. The bits of each global mask array gm1 to gm8 are supplied to the control input of the corresponding switch S1 to S8 on a bit-by-bit basis, the bits being supplied in the order of the blocks to which the local vectors sequentially supplied to the flag analyser 203 correspond. Thus, for example, if the local vector supplied at any time to the flag analyser 203 corresponds to the block at the top left hand corner of the input image, then the top left hand mask bit in each global mask array gm1 to gm8 is supplied to the control input of the corresponding switch S1 to S8.

As each local vector is supplied to the flag analyser 203, the confidence, alias, and ridge flags associated therewith are analysed. Provided all the flags indicate a favourable condition of the local motion vector, then the local motion vector is deemed valid and supplied to a first input of a master switch 204. If any of the flags indicates an unfavourable condition of the local motion vector, then the local motion vector is rejected and the flag analyser 203 outputs the zero motion vector. Thus, as the local motion vectors are successively supplied to the flag analyser 203, the flag analyser successively supplies the valid local motion vectors to the switch 204, and replaces any invalid vectors by the zero motion vector.

As the local motion vectors are successively supplied to the flag analyser 203, the address generator 205 controls the reading of "neighbouring" motion vectors from the local vector array 201. The neighbouring vectors for a given local motion vector are those local motion vectors corresponding to blocks neighbouring the block to which the local motion vector corresponds. In this example, eight neighbouring vectors are used for each local motion vector, the eight vectors corresponding to the eight blocks surrounding the block associated with that local motion vector in an array of 3x3 blocks in the input image. For each local motion vector supplied to the flag analyser 203, therefore, eight neighbouring motion vectors are read out from the local vector array 201 and supplied in parallel to a further flag analyser 206 which performs, for the neighbouring motion vectors, a similar function to the flag analyser 203. Thus, the flag analyser 206 outputs only valid neighbouring vectors (as indicated by the associated flags as previously described), replacing any invalid neighbouring vectors by the zero vector. The eight vectors output by the flag analyser 206 are supplied to the first inputs of respective switches N1 to N8.

As indicated in the figure, the long vector flag LV associated with each local motion vector, and derived by the global vector detector 103 as described earlier, is also supplied to the block vector allocator 105. As an extension of the idea of restricting the application of long vectors during global vector detection, it is preferred that, where the long vector flag LV is set for a given local motion vector, only that local motion vector and the zero motion vector are passed to the motion vector selector 14 for possible selection as an output pixel motion vector. (For the reason given earlier, a long vector (though valid) may not provide a high degree of correlation during the vector selection process, and thus an incorrect neighbouring or global vector might otherwise be selected instead). Thus, neighbouring vectors corresponding to a long local motion vector will not be required in the subsequent processing. Accordingly, the switches N1 to N8 which receive the eight neighbouring (or zero) motion vectors for a given local motion vector are controlled by the long vector flag LV for that local motion vector. If the long vector flag is set, then the switches N1 to N8 switch out the neighbouring vectors supplied to their inputs, with the result that the zero motion vector is supplied to the output of each of the switches N1 to N8. This is achieved by connecting the switch outputs to respective switch inputs which are permanently connected to zero. If the long vector is not set for a given local motion vector, then the eight neighbouring (or zero) motion vectors supplied to the switches N1 to N8 are supplied to their respective outputs. The outputs of the switches N1 to N8 are connected to eight further inputs of the master switch 204 as shown in the figure. Thus, as a given local motion vector (or zero vector) is supplied by the flag analyser 203 to the corresponding input of the switch 204, either the neighbouring vectors for that local vector, or the zero motion vector, are supplied to the eight further inputs of the switch 204.

As indicated above, as a given local motion vector is supplied to the flag analyser 203, corresponding mask bits of the eight global masks gm1 to gm8 are supplied to the control inputs of the switches S1 to S8. In each switch S1 to S8, if the global mask bit currently supplied to the control input is set, ie has the value "1", then the corresponding global vector supplied to the switch input is passed on to the switch output. Thus, for a given local motion vector, only those global vectors g1 to g8 for which the relevant mask bit in the corresponding global mask gm1 to gm8 is set are passed on by the switches S1 to S8 for use in the subsequent vector processing. The effect of this is therefore to restrict the assignment of global motion vectors to blocks at or near areas in the input image where motion within a predetermined tolerance of the global vector was derived as previously described. If the mask bit controlling a switch S1 to S8 is not set, then the zero motion vector (to which one input of each switch S1 to S8 is permanently connected) is supplied to the switch output.

The outputs of the eight switches S1 to S8 are supplied to respective first inputs of a further eight switches P1 to P8 each of which is controlled by the long vector flag for the corresponding local motion vector supplied to the flag analyser 203. The switches P1 to P8 operate in the same way as the switches N1 to N8. Thus, if the long vector flag is set for a particular local motion vector then the zero motion vector, to which one input of each switch P1 to P8 is permanently connected, is supplied to the output of each of the switches P1 to P8. If the long vector flag is not set for that local vector, then any global (or zero) vectors supplied to the inputs of the switches P1 to P8 are supplied to the switch outputs. The outputs of the switches P1 to P8 are connected to eight further inputs to the master switch 204 as shown in the figure. Thus, as a given valid local (or zero) vector is supplied to the upper input of the master switch 204, the corresponding eight neighbouring (or zero) vectors are supplied by the switches N1 to N8 to the next eight inputs of the master switch 204, and any associated global (or zero) vectors are supplied to the lower set of inputs of the master switch 204.

As each local (or zero) motion vector is supplied to the upper input of the master switch 204, the output 207 of the master switch 204 is sequentially connected to each of the switch inputs starting with the upper input and working down to the lowest input (which corresponds to the output of the switch P8). The vectors supplied at the inputs of the master switch 204 are thus sequentially supplied to its output 207. The output 207 of the master switch 204 forms one input to each four comparators 208a, 208b, 208c and 208d. The output 207 is also connected to the input of a vector stack 209 which can store (in this example) up to four vectors at locations indicated schematically as L0 to L3. All locations L0 to L3 are initially occupied by the zero motion vector. The contents of the locations L0 to L3 in the vector stack 209 are supplied to the second inputs of the comparators 208a to 208d respectively. Each of the comparators 208a to 208d thus compares the vector currently supplied on the output 207 of the master switch 204 with the vector stored in one of the locations L0 to L3 respectively of the stack 209. Each comparator 208a to 208d generates an output of "1" if the vectors compared thereby are not equal. If the vectors compared by a comparator 208a to 208d are equal, the output of the comparator is set to zero.

The outputs of the comparators 208a to 208d form respective inputs to an AND gate 210. If all the inputs to the AND gate 210 are set to "1" then the gate 210 generates a "load vector" pulse which is supplied to the vector stack 209. The load vector pulse causes the vector currently supplied on the output 207 of the master switch 204 to be stored in the first of the locations L1 to L3 of the stack which is occupied by the zero vector. Location L0 is permanently occupied by the zero motion vector since the four motion vectors to be supplied to the motion vector selector 16 for each block always include the zero vector. Thus, the first non-zero vector supplied to the master switch output 207 will be stored in location L1. The next vector supplied to the output 207 of the master switch 204 which is non-zero and different to that already stored at location L1 will be stored in location L2, and so on. Since the master switch output 207 is sequentially connected to each of the master switch inputs from the upper input downwards, the resulting effect is that a set of up to four unique vectors is eventually stored in the stack 209, the vectors having been selected, in order of preference, from the zero motion vector (which is always used), a valid local motion vector, any valid neighbouring vectors (unless the corresponding local vector is a long vector), and any global vectors g1 to g8 which have been associated with that local vector (again unless the local vector is a long vector). Further, the global vectors g1 to g8 are in order of decreasing frequency of occurrence of the corresponding local vectors. Thus, global vectors (if selected) will be selected in order of decreasing frequency of occurrence over the input image as a whole.

Once all vectors supplied to the switch 204 have been tested, the vectors stored in locations L0 to L3 are supplied to respective first inputs of four further comparators 211a, 211b, 211c and 211d. The comparators 211a to 211d each compare one of the four vectors with a further magnitude threshold. This magnitude threshold is set to a higher level than that used for detecting long vectors in the global vector detector 103 as described earlier. For example, the threshold could be set at ¾ of the maximum horizontal and vertical vector components. The outputs of the comparators 211a to 211d form respective inputs to an OR gate 212. If the output of any one of the comparators 211a to 211d indicates that the corresponding vector exceeded the magnitude threshold, and is thus a "very long" vector, then an interpolation flag is generated on an output 213 of the OR gate 212. (In the present case, since the long vector flag LV has already been used to switch out the neighbouring and associated global vectors corresponding to a "long" local vector, it could in this case only be a long vector itself which exceeds the higher threshold.) The reason for this further threshold test is that, for very rapidly moving objects, the eye is less able to detect the judder or blurring introduced by non-motion compensated interpolation. These effects can therefore be less subjectively disturbing than motion compensated interpolation using an incorrect vector. Thus, for vectors classified as "very long" vectors, it is preferable to pass only the zero vector to vector selection and to supply the interpolation flag to the later processing stages to indicate that the interpolation process should be modified accordingly.

Accordingly, the interpolation flag on the output 213 of the OR gate 212 is used to control four further switches Q1 to Q4 each of which receives one of the four vectors from the vector stack 209. If the interpolation flag is not set, then the switches Q1 to Q4 simply supply to their outputs the four vectors (indicated as 0, V1, V2, and V3 in the figure) to their outputs. However, if the interpolation flag is set, all four vectors 0, V1, V2 and V3 are effectively set to zero, the switches Q1 to Q4 supplying the zero motion vector (permanently supplied to one input of each switch) to their outputs.

The operation of the block vector allocator 105 described above is repeated for each local vector supplied to its input 200. Thus, the effect of the block vector allocator 105 is to assign to each block of the input image (to which a given local motion vector corresponds) a set of four vectors 0, V1, V2 and V3. The vectors 0, V1, V2 and V3 are supplied on the output 107 of the motion vector reducer 13 to the motion vector selector 14 shown in Figure 2. The interpolation flags supplied on the output 213 of the OR gate 212 in Figure 22 are also output to the motion vector selector 14. As indicated above, if the interpolation flag is set, the corresponding vectors 0, V1, V2 and V3 will all be the zero motion vector.

In the motion vector selector 14, a single motion vector is selected for each pixel of a block of an output field corresponding to a block of the original input field. Thus, for each output pixel, the four vectors 0, V1, V2, V3 supplied for the corresponding input field are considered. If the interpolation flag output by the block vector allocator 105 is set for the corresponding block, then the motion vector selector receives only the zero motion vector and therefore supplies the zero motion vector to the subsequent processing stages for all pixels in that block. Otherwise, for each output pixel, each of the vectors 0, V1, V2 and V3 is tested to determine which corresponds to the "best" vector for that pixel. This test is indicated schematically in Figure 23 for the zero motion vector 220 and a non-zero motion vector 221. The output pixel 222 to be produced for an output field 223 is considered to be positioned at its correct temporal position along the motion vector 220, 221 under test. The motion vector under test then "points" to groups of pixels, or "test blocks", 224 in each of the original pair of input fields n and n+1. The test blocks 224 may, for example, be taken to comprise a group of 5x5 pixels around the pixel position pointed to by the motion vector under test. The degree of correlation between the two test blocks 224 pointed to by a motion vector under test is then assessed, for example by summing the absolute difference values between corresponding pixels in the two test blocks. The motion vector which provides the best correlation is then selected as the "best" motion vector for that output pixel and is supplied by the vector selector 14 to the subsequent processing stages. The value of the output pixel 222 is then interpolated in the interpolator 8 (Figure 2) from the original two input fields supplied thereto. The output pixel motion vector is used to identify appropriate groups of pixels (for example, patches of 8x8 pixels centred on the pixels pointed to by the motion vector) in the two input fields which are then combined to produce the output pixel, the input pixels being combined in relative proportions depending on the temporal position of the output field relative to the input fields. However, if the interpolation flag (supplied by the motion vector selector 14 to the interpolator 8) is set, then for the reasons indicated earlier, the output pixel is produced in the interpolator without motion compensated interpolation. That is to say, the output pixel motion vector is the zero motion vector, and temporal weighting of the two input field is not used. No temporal weighting is used in this case since if the zero vector is used to produce a given area of the image over a substantial number of output fields, then, if temporal weighting were used, a beating effect may be seen as the temporal offset of the output fields relative to the input field pairs varies periodically. This beating effect may arise because the zero motion vector is being used instead of a "very long" vector which has been rejected in the block vector allocator (Figure 22), and thus clearly inappropriate portions of the input fields may be combined to produce the output pixel. The effect of the temporal weighting may then be clearly visible in the output fields.

CLAIMS

1. Motion compensated video signal processing apparatus in which motion vectors are generated to represent image motion between a pair of images of an input video signal, the apparatus comprising:

means for deriving local motion vectors representing the motion of the content of respective blocks of one image of the pair between that image and the other image of the pair;

means for deriving from the local motion vectors a set of global motion vectors comprising a plurality of unique motion vectors selected from the most common of the local motion vectors;

association means for establishing an association between each global vector and blocks of the said image, the association means comprising means for comparing each global motion vector with each local motion vector, the said association being established between a global motion vector and a given block if the global motion vector is within a predetermined tolerance of the local motion vector for that block; and

means for assigning to each said block a plurality of motion vectors selected from the zero motion vector, the local motion vector for that block, the local motion vectors for the blocks neighbouring that block in the input image, and any global motion vectors for which the said association with that block has been established.

2. Apparatus as claimed in claim 1, wherein the association means includes means for generating for each global vector a global mask array having an array entry for each said block of the input image, wherein each array entry indicates whether or not the said association has been established for the corresponding block.

3. Apparatus as claimed in claim 2, wherein the association means includes first-pass mask adjustment means for determining, for each entry in each global mask array, the number of entries which indicate that the said association has been established in a group of entries comprising that entry and the entries corresponding to the blocks neighbouring the block to which that entry corresponds, comparing the said number with a predetermined threshold, and, if the number is less

than the threshold and the entry indicates that the said association has been established, resetting the entry to indicate that the association has not been established, and, if the number is greater than or equal to the threshold and the entry indicates that the said association has not been established, resetting the entry to indicate that the association has been established.

4. Apparatus as claimed in claim 3, wherein the association means includes additional-pass mask adjustment means for repeating the operation of the first-pass mask adjustment means at least once with the predetermined threshold set to a lower level than that for the first-pass mask adjustment.

5. Apparatus as claimed in claim 4, wherein the additional-pass mask adjustment means is arranged to repeat the operation at least twice.

6. Apparatus as claimed in claim 4, wherein the additional-pass mask adjustment means is arranged to repeat the operation at least three times.

7. Apparatus as claimed in any one of claims 3 to 6, wherein the mask adjustment means is arranged to determine the said number of entries which indicate that the said association has been established in a group of nine entries corresponding to a 3x3 group of blocks in the input image.

8. Apparatus as claimed in any preceding claim, wherein the means for deriving global motion vectors is arranged to select a local motion vector as a global motion vector only if the number of local motion vectors having the value of that local motion vector exceeds a predetermined threshold number.

9. Apparatus as claimed in any preceding claim, wherein the means for deriving global motion vectors is arranged to select a local motion vector as a global motion vector only if the magnitude of the local motion vector is less than or equal to a predetermined magnitude threshold.

10. Apparatus as claimed in claim 9, wherein the means for assigning a plurality of motion vectors to each said block is arranged to assign only the zero motion vector and the corresponding local motion vector to any block for which the magnitude of the corresponding local motion vector exceeds the said predetermined magnitude threshold.

11. Apparatus as claimed in any preceding claim, wherein the means for deriving local motion vectors comprises:

means for comparing a search block of one of the pair of input images with a search area, which is larger than the search block, in the other of the pair of input images, to generate a correlation surface, the correlation surface comprising an array of correlation values representing correlation between the search block and the search area;

means for detecting a maximum correlation value in the correlation surface;

means for generating a local motion vector in dependence on the position of the maximum correlation value in the correlation surface; and

ridge detecting means for detecting correlation values in the correlation surface within a threshold amount of the maximum correlation value, indicative of a ridge correlation maximum.

12. Apparatus as claimed in claim 11, wherein the means for deriving local motion vectors includes:

means for comparing a local motion vector under test, derived from a correlation surface in which a ridge correlation maximum was detected, with a further local motion vector generated from the pair of input images;

means for detecting whether corresponding components of the further local motion vector and the local motion vector under test are substantially identical;

means for detecting whether the further local motion vector was derived from a correlation surface in which a ridge correlation maximum was detected; and

means, responsive to a detection that the further local motion vector was not derived from a correlation surface in which a ridge correlation maximum was detected, for replacing a component of the local motion vector under test in the ridge direction by the corresponding component of the further local motion vector.

13. Apparatus as claimed in any preceding claim, wherein the means for deriving local motion vectors comprises:

means for comparing a search block of one of the pair of input images with a search area, which is larger than the search block, in the other of the pair of input images, to generate a correlation surface, the correlation surface comprising an array of correlation values representing correlation between the search block and the search area;

means for detecting a first point of maximum correlation from the correlation surface;

means for generating a local motion vector in dependence on the first correlation value;

means for detecting a second point of maximum correlation from the correlation surface apart from an excluded region surrounding the first point of maximum correlation;

means for detecting whether the second point of maximum correlation is adjacent to the excluded region of the correlation surface; and

means for setting an alias flag, associated with the local motion vector, to indicate that the local motion vector is potentially aliased if the second point of maximum correlation is not adjacent to the excluded region of the correlation surface.

14. Apparatus as claimed in claim 13, including:

means for comparing a local motion vector under test, for which the associated alias flag is set, with a predetermined group of other local motion vectors generated from the pair of images; and

means, responsive to a detection that the magnitude of the local motion vector under test is within a predetermined difference threshold of the magnitude of at least one of the predetermined group of local motion vectors for which the corresponding alias flag is not set, for resetting the alias flag associated with the local motion vector under test.

15. Motion compensated video signal processing apparatus substantially as hereinbefore described with reference to the accompanying drawings.

16. A motion compensated television standards converter comprising apparatus as claimed in any preceding claim.

17. A method of motion compensated video signal processing in which motion vectors are generated to represent image motion between a pair of images of an input video signal, the method comprising:

deriving local motion vectors representing the motion of the content of respective blocks of one image of the pair between that image and the other image of the pair;

deriving from the local motion vectors a set of global motion vectors comprising a plurality of unique motion vectors selected from the most common of the local motion vectors;

establishing an association between each global vector and blocks of the said image by comparing each global motion vector with each local motion vector, the said association being established between a global motion vector and a given block if the global motion vector is within a predetermined tolerance of the local motion vector for that block; and

assigning to each said block a plurality of motion vectors selected from the zero motion vector, the local motion vector for that block, the local motion vectors for the blocks neighbouring that block in the input image, and any global motion vectors for which the said association with that block has been established.

18. A method of motion compensated image processing as claimed in claim 17, which method is carried out using apparatus as claimed in any one of claims 1 to 16.

19. A method of motion compensated image processing which method is substantially as hereinbefore described with reference to the accompanying drawings.

Patents Act 1977
Examiner's report to the Comptroller under Section 17 (The Search Report)

45

Application number

GB 9307442.5

Relevant Technical fields

(i) UK Cl (Edition L) H4F (FGM)

(ii) Int Cl (Edition 5) H04N (5/14) G06F (15/70)

Search Examiner

M K REES

Databases (see over)

(i) UK Patent Office

(ii) ONLINE DATABASES: WPI, INSPEC.

Date of Search

27 JULY 1993

Documents considered relevant following a search in respect of claims 1 TO 19

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
A	GB 2249899 A (SONY) see abstract	1, 17
A	EP 0498938 A2 (TELENORMA GMBH) see abstract; Figure 1	1, 17

Category	Identity of document and relevant passages	Relevant to claim(s)

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family, corresponding document.

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).